mi COMPUTER

32

CURSO PRACTICO DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR



150 ptas.

Editorial Delta, S.A.

MI COMPUTER CURSO PRACTICO DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR

Publicado por Editorial Delta, S.A., Barcelona, y comercializado en exclusiva por Distribuidora Olimpia, S.A., Barcelona

Volumen III - Fascículo 32

Director:	José Mas Godayol
Director editorial:	Gerardo Romero
Jefe de redacción:	Pablo Parra
Coordinación editorial:	Jaime Mardones
Asesor técnico:	Francisco Martín Jesús Nebra

Redactores y colaboradores: G. Jefferson, R. Ford, S. Tarditti. A. Cuevas, F. Blasco

Para la edición inglesa: R. Pawson (editor), D. Tebbutt (consultant editor), C. Cooper (executive editor), D. Whelan (art editor), Bunch Partworks Ltd. (proyecto y realización)

Realización gráfica: Luis F. Balaguer

Redacción y administración:
Paseo de Gracia, 88, 5.º, Barcelona-8
Tels. (93) 215 10 32 / (93) 215 10 50 - Télex 97848 EDLTE

MI COMPUTER, *Curso práctico del ordenador personal, el micro y el miniordenador,* se publica en forma de 96 fascículos de aparición semanal, encuadernables en ocho volúmenes. Cada fascículo consta de 20 páginas interiores y sus correspondientes cubiertas. Con el fascículo que completa cada uno de los volúmenes, se ponen a la venta las tapas para su encuadernación.
El editor se reserva el derecho de modificar el precio de venta del fascículo en el transcurso de la obra, si las circunstancias del mercado así lo exigieran.

ISBN: 84-85822-83-8 (fascículo) 84-85822-94-3 (tomo 3)
84-85822-82-X (obra completa)
Depósito Legal: B. 52/1984

Fotocomposición: Tecfa, S.A., Pedro IV, 160, Barcelona-5
Impresión: Cayfosa, Santa Perpètua de Mogoda (Barcelona) 228408
Impreso en España - Printed in Spain - Agosto 1984

Editorial Delta, S.A., garantiza la publicación de todos los fascículos que componen esta obra.

Distribuye para España: Marco Ibérica, Distribución de Ediciones, S.A., Carretera de Irún, km 13,350. Variante de Fuencarral, Madrid-34.

Distribuye para Argentina: Viscontea Distribuidora, S.C.A., La Rioja 1134/56, Buenos Aires.

Distribuye para Colombia: Distribuidoras Unidas, Ltda., Transversal 93, n.º 52-03, Bogotá D.E.

Distribuye para México: Distribuidora Intermex, S.A., Lucio Blanco, n.º 435, Col. San Juan Tlihuaca, Azcapotzalco, 02400, México D.F.

Distribuye para Venezuela: Distribuidora Continental, S.A., Ferrenquín a Cruz de Candelaria, 178, Caracas, y todas sus sucursales en el interior del país.

Pida a su proveedor habitual que le reserve un ejemplar de MI COMPUTER. Comprando su fascículo todas las semanas y en el mismo quiosco o librería, Vd. conseguirá un servicio más rápido, pues nos permite realizar la distribución a los puntos de venta con la mayor precisión.

Servicio de suscripciones y atrasados (sólo para España)

Las condiciones de suscripción a la obra completa (96 fascículos más las tapas, guardas y transferibles para la confección de los 8 volúmenes) son las siguientes:

a) Un pago único anticipado de 16 690 ptas. o bien 8 pagos trimestrales anticipados y consecutivos de 2 087 ptas. (sin gastos de envío).

b) Los pagos pueden hacerse efectivos mediante ingreso en la cuenta 3371872 de la Caja Postal de Ahorros y remitiendo a continuación el resguardo o su fotocopia a Distribuidora Olimpia (Paseo de Gracia, 88, 5.º, Barcelona-8), o también con talón bancario remitido a la misma dirección.

c) Se realizará un envío cada 12 semanas, compuesto de 12 fascículos y las tapas para encuadernarlos.

Los fascículos atrasados pueden adquirirse en el quiosco o librería habitual. También pueden recibirse por correo, con incremento del coste de envío, haciendo llegar su importe a Distribuidora Olimpia, en la forma establecida en el apartado b).

Para cualquier aclaración, telefonear al (93) 215 75 21.

No se efectúan envíos contra reembolso.



Estrella del futuro

Este ordenador norteamericano incorpora muchas de las configuraciones estándar del avanzado MSX, lo que, unido a su capacidad de ampliación, le augura un brillante porvenir

Para entender por qué los distintos ordenadores personales son tan singulares en cuanto a diseño necesitamos conocer cómo se desarrolló la industria del microordenador. Los primeros microprocesadores que causaron un impacto en el mercado del ordenador personal fueron el 8080 de Intel y, presumiblemente, el 6800 de Motorola. Los conjuntos de instrucciones de estos procesadores se fijaron desde el comienzo. El problema de la compatibilidad surgió por la versatilidad de los procesadores. Por ejemplo, enviar un byte de datos a una puerta de salida implicaba exactamente la misma instrucción, cualquiera que fuera el ordenador, siempre y cuando se utilizara el mismo procesador. Pero la salida podía tener cualquier dirección entre centenares o millares de direcciones. Muchos de los creadores de los primeros micros personales los construyeron en los garajes o en los patios de sus casas, por lo cual, como es natural, no tenían ningún tipo de acuerdo entre ellos acerca de en qué lugar del mapa de memoria deberían localizarse las puertas de entrada o salida. El fabricante A podía escoger que la puerta para salida a impresora en paralelo estuviera localizada en la dirección 255, mientras que el fabricante B podía escoger para lo mismo la dirección 254.

Esta situación ya era negativa de por sí, pero con el advenimiento de los gráficos en pantalla controlados por chips controladores CRT especiales, y los efectos de sonido controlados por chips exclusivos para sonido, la situación se hizo todavía más caótica. Y así se garantizó que todo programa que aprovechara al máximo las capacidades especiales de cualquier ordenador no se pudiera ejecutar en otro ordenador como no fuese a costa de un considerable trabajo de reescritura.

De haber habido alguna empresa que estuviera en una posición de suficiente poder como para forzar un estándar desde el principio, la situación habría sido totalmente distinta. Pero no fue ése el caso. En los primeros tiempos de los ordenadores personales había numerosos pequeños fabricantes, cada uno de ellos con un estilo y un estándar propios y diferentes. No se trataba tan sólo de que las máquinas eran distintas las unas de las otras física y electrónicamente, sino que hasta los lenguajes de programación que se proporcionaban con ellas eran incompatibles con otras máquinas. Desde el principio, el BASIC jamás respondió a un estándar. Hasta la década de los setenta, como mínimo, la comunidad informática profesional no se tomaba muy en serio el BASIC, al que consideraban un lenguaje estrictamente para principiantes.

Desde finales de los setenta a principios de los ochenta los microordenadores se desarrollaron a ritmo acelerado. Diseños pioneros, como el Apple, comenzaron a incorporar refinamientos tales como gráficos y color sofisticados. Para que estas innovaciones se pudieran utilizar, los fabricantes tuvieron que desarrollar sus propias versiones de BASIC, y así empezó la diversificación del lenguaje.

El precio de esta proliferación de versiones no es tanto la disposición de mayores opciones por parte del consumidor, sino más bien la frustración que

Tony Duncan-Smith

El camino hacia el éxito
El estándar MSX es la ruta japonesa hacia los mercados mundiales de ordenadores. De conseguir el éxito, Japón podría llegar a dominar el mercado del ordenador, tal como consiguiera hacerlo en los campos de la alta fidelidad y las cámaras fotográficas

experimentan por igual propietarios, fabricantes y escritores de software.

Si usted fuera un fabricante de ordenadores con un nuevo producto que lanzar al mercado, sabría que prácticamente no habrá software para su máquina hasta que se haya establecido un sustancial número de usuarios. Con poco o nada de software compatible con su ordenador, será muy consciente de que las previsiones de ventas de su máquina son limitadas y que esto puede poner en peligro la inversión que supone el desarrollo del producto.

Si se ganara la vida escribiendo software comercial, sus ventas estarían limitadas (en el mejor de los casos) a la cantidad de personas que posean el modelo de ordenador para el cual ha escrito el programa. Supongamos que ha creado un juego de aventuras que se llama *Los calabozos de Rathbone*, todo completo, con una Presencia Sin Rostro, un sádico jefe de calabozos e innumerables escondrijos siniestros para atrapar a los incautos. Los investigadores de mercado podrían afirmar que el potencial de ventas del producto sería inmenso si pudiera sacarlo al mercado al precio de 1 200 pesetas y vender al menos 65 000 ejemplares. Lamentablemente, las previsiones del producto son tan altas que sería improbable que el número de usuarios de Spectrum por sí solo generara la cifra de ventas requerida, y se necesitaría al menos una versión más. El costo que supondría la producción de nuevas versiones para, pongamos por caso, el Oric y el BBC Micro elevaría el precio de la unidad a 1 500 pesetas y, de este modo, las ventas caerían por debajo de un nivel económicamente viable. Éste es el dilema que frustra a muchos posibles escritores de software.

La industria informática no ha ignorado el problema de la incompatibilidad de software. El individualista mundo occidental no se muestra muy dispuesto a la estandarización, pero quienes desarrollan ordenadores en el Lejano Oriente prefieren que las cosas sean sistemáticas y estandarizadas.

ASCII/Microsoft está intentando reemplazar el caos por el orden. La empresa es el resultado de una fusión en la que están comprometidas la American Microsoft Corporation y ASCII, una exitosa editorial japonesa que tiene en su haber una serie de populares publicaciones. Como actividad complementaria, ASCII también edita software comercial, y cuando Microsoft quiso introducirse en el impenetrable mercado japonés, ASCII pareció la firma idónea a la cual dirigirse para una empresa conjunta. El razonamiento era muy válido, ya que Microsoft poseía la necesaria experiencia técnica y ASCII la experiencia en comercialización.

American Microsoft Corporation ha conseguido su prestigio en virtud de lo que más se parece a un estándar del BASIC, el MBASIC Microsoft, que han adoptado muchos fabricantes de ordenadores en todo el mundo. Pero, así y todo, no se podía garantizar que cualquier programa en MBASIC se pudiera ejecutar en todos los ordenadores que utilizaran ese lenguaje siempre que hubiera de por medio alguna configuración especial, simplemente debido a la carencia de una compatibilidad de hardware.

El BASIC Microsoft se vendió con gran éxito a numerosos fabricantes japoneses de ordenadores, a través de las oficinas de ASCII/Microsoft. Pero aun así esto no solucionó el problema de la compatibilidad de hardware y software. La solución de ASCII/Microsoft consistió en crear un estándar, en colaboración con los principales fabricantes japoneses, con el deseo de que se lo reconociera internacionalmente como tal. Al resultado final que obtuvieron se lo denomina Estándar MSX. La especificación incluye exigencias básicas de hardware (basadas alrededor del microprocesador Z80 y otros chips fijos), así como un lenguaje estandarizado.

Las especificaciones MSX

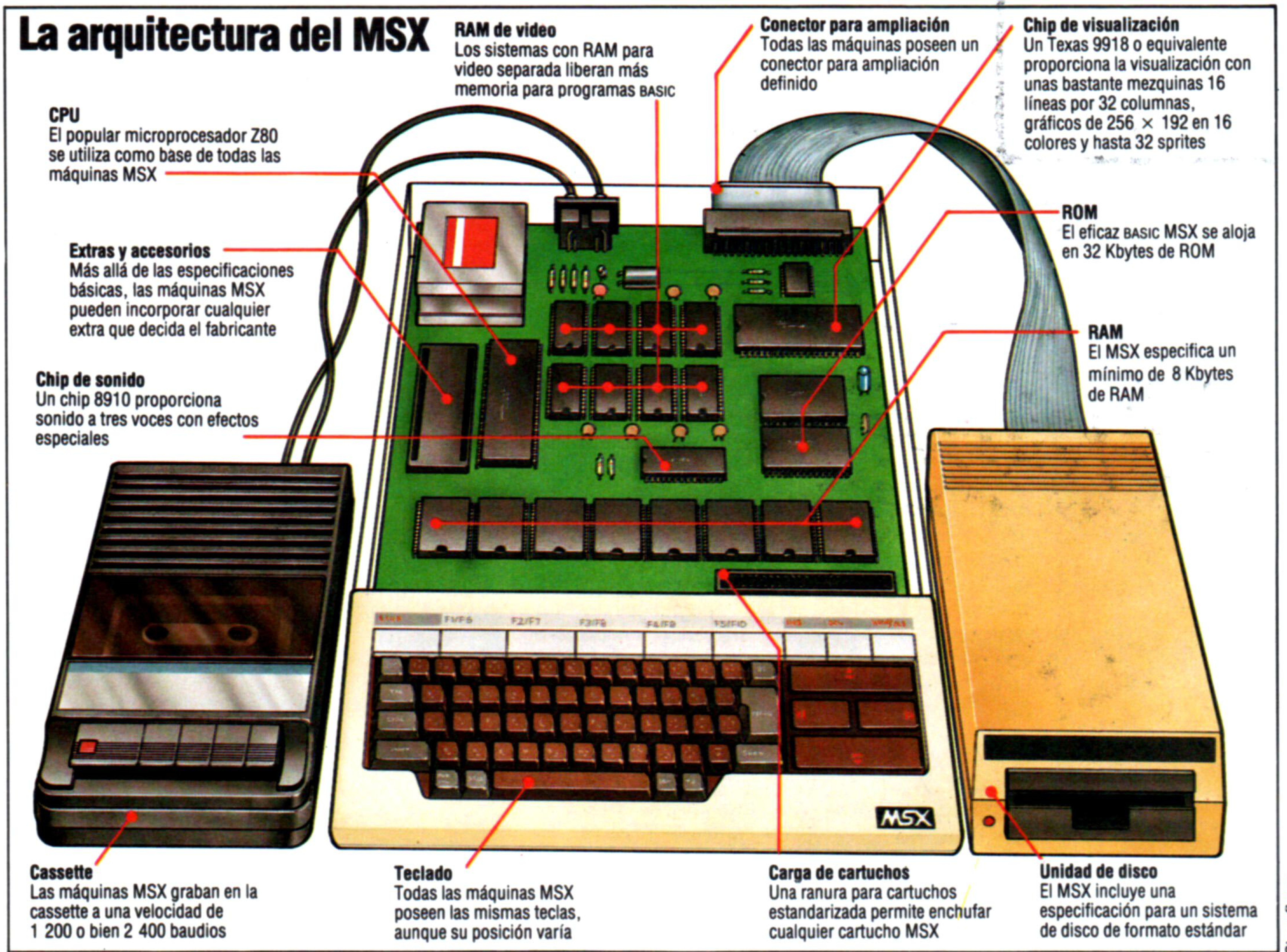
El BASIC MSX es muy similar al MBASIC de Microsoft, pero incluye varias significativas mejoras para sacar partido de las capacidades para gráficos y sonido actuales. Entre las sentencias incorporadas se incluyen SCREEN, para especificar la modalidad de pantalla, el tamaño de los sprites, el "tecleo" de teclas, la velocidad baudio de la cassette y las opciones de impresora; LOCATE, para el posicionamiento de caracteres en la pantalla; COLOUR, para seleccionar uno de los 16 colores de fondo y primer plano; PUT SPRITE, para establecer las características de los sprites; CIRCLE, para trazar círculos y elipses; DRAW, para dibujar figuras; LINE, para trazar líneas entre coordenadas especificadas, y PAINT, para rellenar figuras con un determinado color. También se proporciona una sentencia KEY para asignarles cadenas a las teclas de función. Hay otras sentencias para colocar valores en la RAM de video (VPOKE), escribir valores en los registros del chip de efectos sonoros (SOUND) y para controlar el motor de la cassette (MOTOR).

El MSX, sin embargo, implica algo más que un software estandarizado. La CPU se especifica como un Z80 que trabaja a 3,58 MHz. Debe haber al menos 32 Kbytes de ROM para almacenar el software MSX. Además, debe haber al menos ocho Kbytes de RAM. No existe límite máximo en cuanto a la cantidad de ROM y RAM permitida. Un ordenador MSX ha de incorporar un chip controlador de video TMS9918A de Texas Instruments (o equivalente) y un AY-3-8910, chip generador de sonido a tres voces. La salida de video debe ser capaz de visualizar o 32 columnas por 24 líneas, o bien 40 columnas por 24 líneas. En la actualidad, en las especificaciones no está prevista una visualización a 80 columnas. Se requiere una resolución de 256 por 192 pixels.

La interface para cassette se ha determinado, siendo las cassettes el medio físico principal para el almacenamiento de programas y de datos. Debe utilizar el sistema de codificación FSK a 1 200-2 400 bits por segundo. El teclado, reflejando el linaje japonés del MSX, no sólo tiene un trazado estándar, incluyendo teclas de función, sino que proporciona las dos categorías de signos silábicos *kana*, escritos en columnas verticales: *katakana* (fragmentos de ideogramas) e *hiragana* (abreviaciones de ideogramas, trazadas en cursiva); además suministra caracteres estandarizados para gráficos y, opcionalmente, un sistema de escritura con caracteres chinos.

Se provee software enchufable a través de una ranura estándar para cartuchos de ROM y hay un bus de entrada-salida, también estándar, de 50 patillas. Hay incluso una puerta estandarizada para dos palancas de mando.

Los formatos de disco están, asimismo, estandarizados, al igual que el sistema operativo en disco,



el MSX-DOS. Funcionalmente es equivalente al MS-DOS y permite leer archivos de datos de éste. También se afirma que es compatible con el enormemente popular sistema operativo en disco CP/M 2.2. También se han determinado formatos para los discos flexibles de 3 ½ pulgadas (Sony), de 5 ¼ pulgadas y de 8 pulgadas.

Todo esto significa que, de haberse ceñido a la normativa, cualquier programa escrito para una máquina MSX, almacenado en cualquier disco, está garantizado para su ejecución en cualquier otro ordenador MSX y será capaz de aprovechar por completo sus capacidades para gráficos y para sonido. Las ventajas, tanto para el fabricante como para el consumidor, son obvias.

Pero el sistema MSX tiene algunas desventajas. La primera de ellas es que todo "estándar" no conseguirá sacar partido de las innovaciones que se produzcan.

El segundo inconveniente reside en que los microprocesadores de ocho bits, de los cuales el de más éxito es el Z80, tienen una vida limitada, son incapaces de direccionar más de 64 Kbytes de memoria principal directamente, y tampoco pueden manipular datos cuyo valor sea superior a 256. Desde este punto de vista, se podría considerar que el estándar MSX es un intento por quemar el último cartucho para prolongar la vida del Z80, que está destinado a disfrutar tan sólo de un éxito a corto plazo en el mercado.

El MSX tal vez proporcione un indicador para el futuro. Es difícil suponer seriamente que el mercado del microordenador vaya a estar dominado por los procesadores de ocho bits dentro de cinco o diez años. Si el estándar MSX consigue algo, es probable que este algo sea recordarles a quienes tienen a su cargo el desarrollo y perfeccionamiento de los ordenadores que en toda innovación en el campo de la informática, la estandarización debe venir lo más temprano posible. El MSX puede constituir una oportuna aportación para aquellos fabricantes que actualmente tienen productos basados en el Z80 a la espera de ser vendidos; es difícil predecir si producirá algún impacto a largo plazo, como no sea convencer al resto del mundo de que la estandarización es importante. En el campo de los ordenadores de 16 bits, IBM ha demostrado que "querer es poder" con su ordenador personal, que se ha convertido en un estándar de facto. ¿Será capaz el estándar MSX de hacer lo mismo por el micro personal de ocho bits? Hasta el momento el MSX está respaldado por un total de 16 fabricantes, incluyendo a Yamaha, JVC, Hitachi, Sony, Sanyo, National, Pioneer, Canon, Fujitsu y Mitsubishi, de Japón, la empresa norteamericana Spectravideo y la coreana Daewoo. Ningún fabricante británico se les ha unido hasta la fecha. Sólo el tiempo y el mercado demostrarán si el mundo necesita más de lo mismo, o bien el tipo de innovaciones individualistas como las aportadas por sir Clive Sinclair.

Convención de diseño
Para lograr que los programas y los accesorios sean compatibles con todos los sistemas MSX, el diseño de un micro MSX sigue reglas estrictas. Una vez la máquina ha alcanzado las especificaciones básicas que reflejamos aquí, los diseñadores pueden agregar sus propios extras especiales

El eficaz transistor

En informática, el más significativo de los componentes electrónicos es el transistor. ¿Cuál es su función?

Los transistores tienen dos misiones esenciales: actuar como amplificador de una señal, o encender y apagar una corriente bajo el control de otra corriente. Es esta habilidad para actuar como un interruptor electrónico lo que los hace útiles en informática. Agrupándolos entre sí, uno puede construir circuitos que almacenen patrones encendido-apagado (*on-off*) que el ordenador puede tratar como números binarios. Otros circuitos son puertas lógicas que permiten sumar secuencias encendido-apagado entre sí, etc.

Si usted construye las puertas lógicas descritas en la página contigua, observará que los ordenadores construidos enteramente a partir de transistores serían máquinas sumamente grandes y caras, tal como lo fueron en realidad alguna vez. Los pequeños ordenadores, de precio razonable, todavía necesitan otro refinamiento: los circuitos integrados. Éstos son circuitos predefinidos con cientos de transistores grabados en un diminuto chip de silicio, encerrados en una cubierta de plástico negro. Usted puede comprar circuitos integrados simples (denominados *chips TTL*), que realizarán satisfactoriamente la mayoría de las tareas que se les encomiende. Cuatro puertas AND en un chip pueden adquirirse a un precio bastante asequible.

La práctica de colocar cada vez más circuitos en un único chip se conoce con las siglas VLSI (*Very Large Scale Integration*: integración a escala muy grande). Los chips VLSI poseen muchos miles de componentes comprimidos en ellos y pueden realizar tareas muy complicadas. El microprocesador es un chip de esta clase. Y también lo son los chips que generan la visualización en televisión, controlan las interfaces y producen la gama de efectos de sonido que pueden proporcionar muchas máquinas. Los principios son los mismos que los utilizados para tres puertas lógicas simples.

Los chips VLSI se construyen mediante tecnologías diferentes, según la relación que se desee obtener entre economía, rendimiento y consumo de energía eléctrica. El tipo que incorporan la mayoría de los ordenadores son los chips MOS (metal-óxido-silicio), mientras que muchos portátiles a pilas utilizan chips CMOS (metal complementario-óxido-silicio), que son más lentos pero necesitan muy poca alimentación eléctrica.

En el próximo capítulo de *Bricolaje* construiremos un sumador incompleto, basado en el circuito de la página 513 de la serie *Ciencia informática*.

El transistor como amplificador

Un transistor tiene tres partes, denominadas base, colector y emisor, con tres cables conectados a cada una de ellas. La base suele utilizarse para controlar una corriente que fluya desde el colector al emisor. Si se aplica una corriente a la base, otra puede fluir del colector hacia el emisor. Una corriente cambiante aplicada a la base hace que la resistencia del trayecto del colector al emisor varíe al unísono. Alimentando en el colector una gran corriente y variándola con una pequeña a través de la base, el emisor produce una señal grande que varía exactamente como la señal pequeña. El transistor se ha utilizado para amplificar la señal pequeña



El transistor como interruptor

La conmutación mediante un transistor supone la utilización a fondo de las propiedades de la base, del colector y del emisor. Un transistor sólo puede dejar pasar una determinada cantidad de corriente a través de su trayecto del colector al emisor. Tiene un punto de saturación en el cual la cantidad de corriente que fluye a través de ese trayecto ya no se ve afectada por las pequeñas variaciones en la cantidad de corriente que fluya a través de la base



CORRIENTE A BASE

LA RESISTENCIA CAE

Creación de puertas lógicas

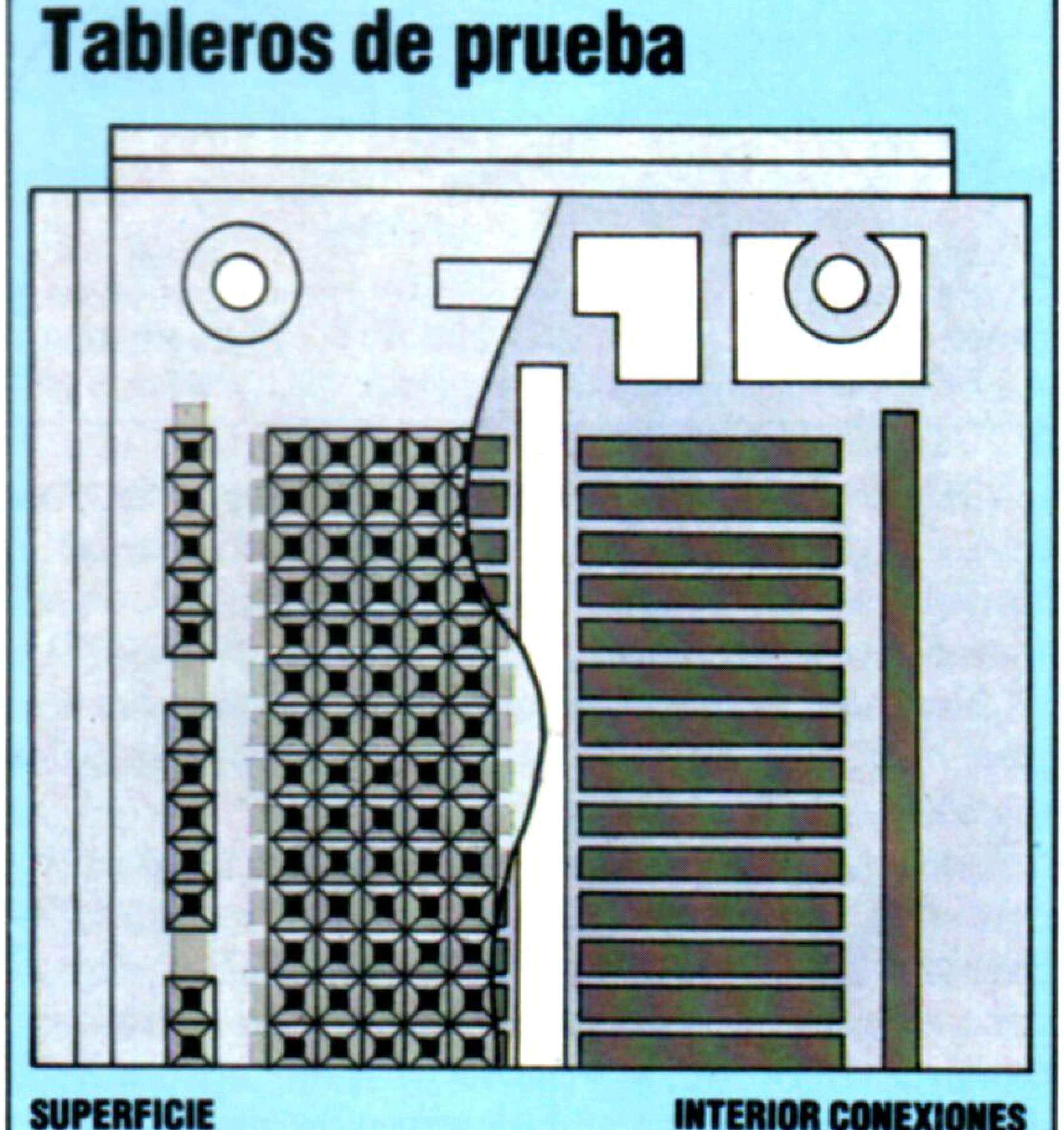
Estos sencillos circuitos (derecha) ilustran la forma en que se puede emplear la capacidad de conmutación del transistor para construir puertas lógicas. Usted mismo puede crearlas de una en una utilizando el mismo conjunto de componentes y un tablero de prueba. Es importante notar que la conmutación real está "transistorizada", o sea, no posee partes móviles. En estos ejemplos, las entradas a los circuitos son botones con indicadores LED (diodos emisores de luz). En un ordenador, las entradas a tales circuitos serían las salidas de otros. Después de haber construido y comprendido estas puertas, quizá se anime a construir un circuito más complejo alimentando la salida de una puerta a otra

Materiales

1 tablero de prueba
2 transistores BC109
2 LED rojos
1 LED verde
3 resist. de 500 ohm
2 resist. de 15 K ohm
2 interruptores
1 pila de 9 voltios
1 conector para pilas
Trozos cortos de cable

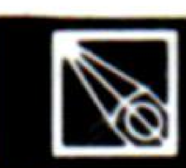
BC 109
COLECTOR
BASE
EMISOR
CARA INFERIOR

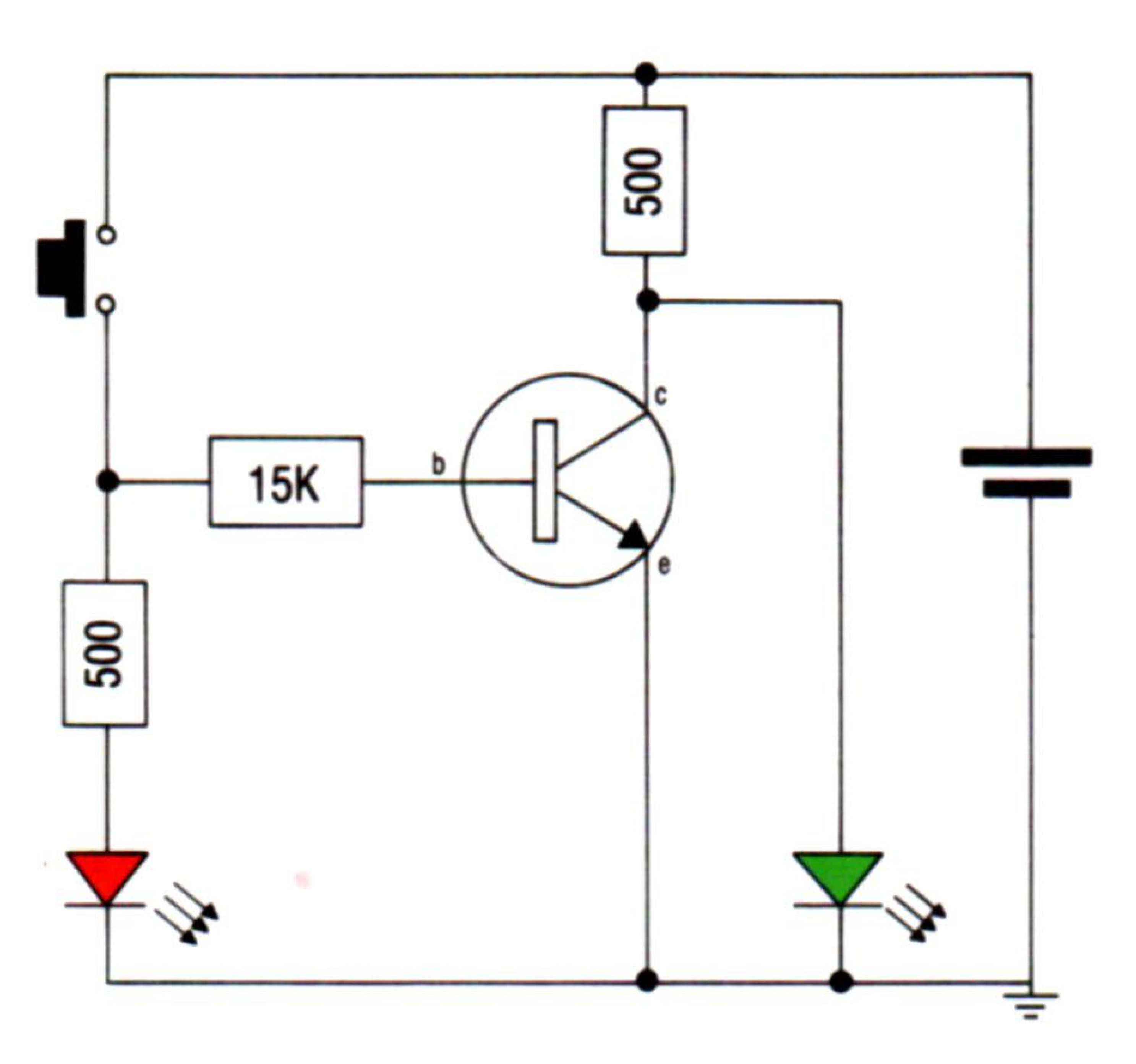
Tableros de prueba



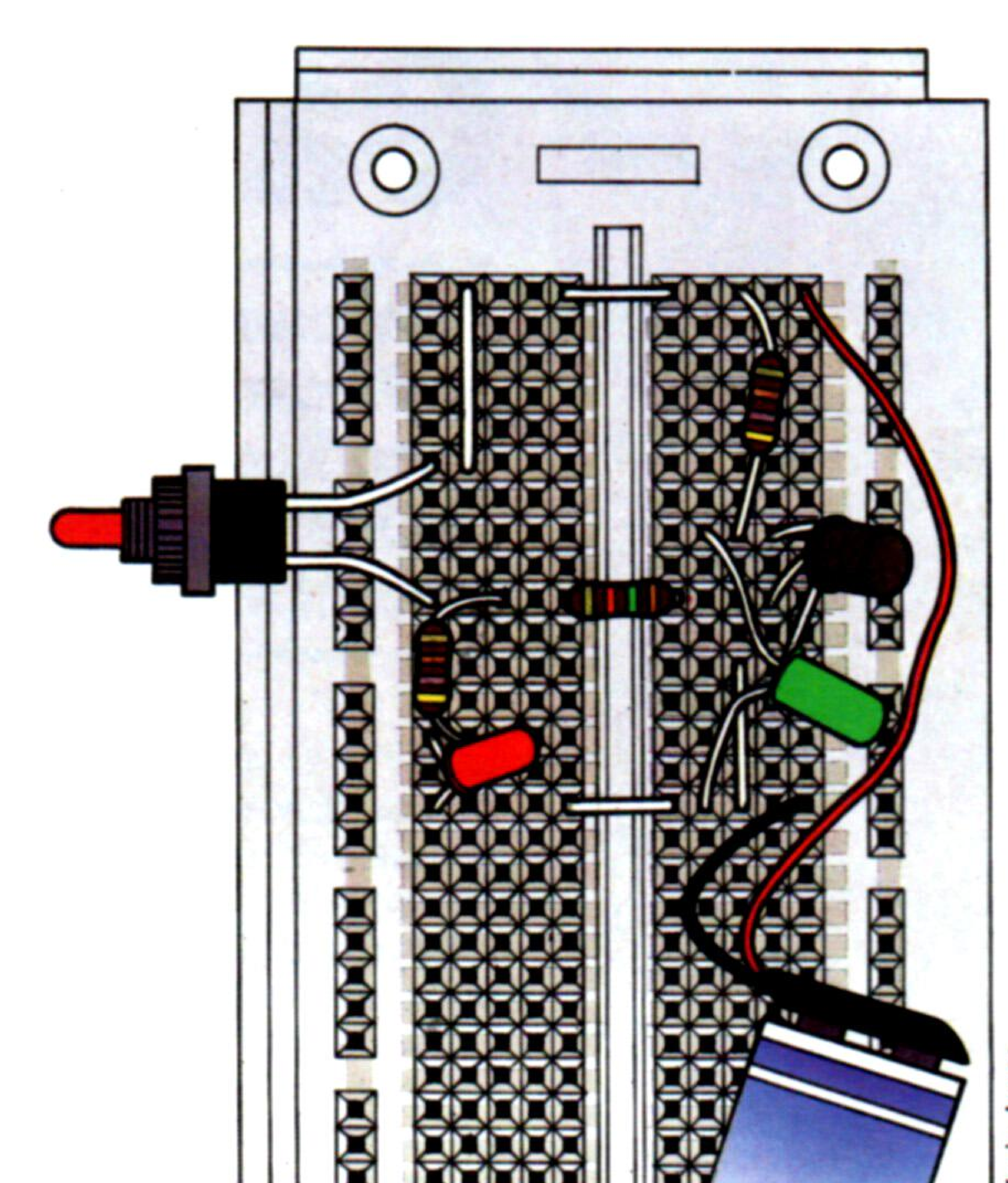
Kevin Jones

Una placa de circuito impreso como la de la figura proporciona una forma sencilla de experimentar con circuitos como los que mostramos sin perder tiempo ni trabajo en soldar los componentes. Consiste en una base siempre recuperable, sobre la cual se enchufan con fijeza los componentes. Las pinzas metálicas para los componentes actúan como conductores, de modo que cada grupo de cinco agujeros está conectado eléctricamente. Con esta matriz es fácil trasladar los diseños de circuitos simples al tablero, utilizando trozos cortos de cable para conectar grupos separados de agujeros. El tablero ilustrado aquí es más grande que el que usted necesita ahora, pero le será útil para futuros y más ambiciosos proyectos

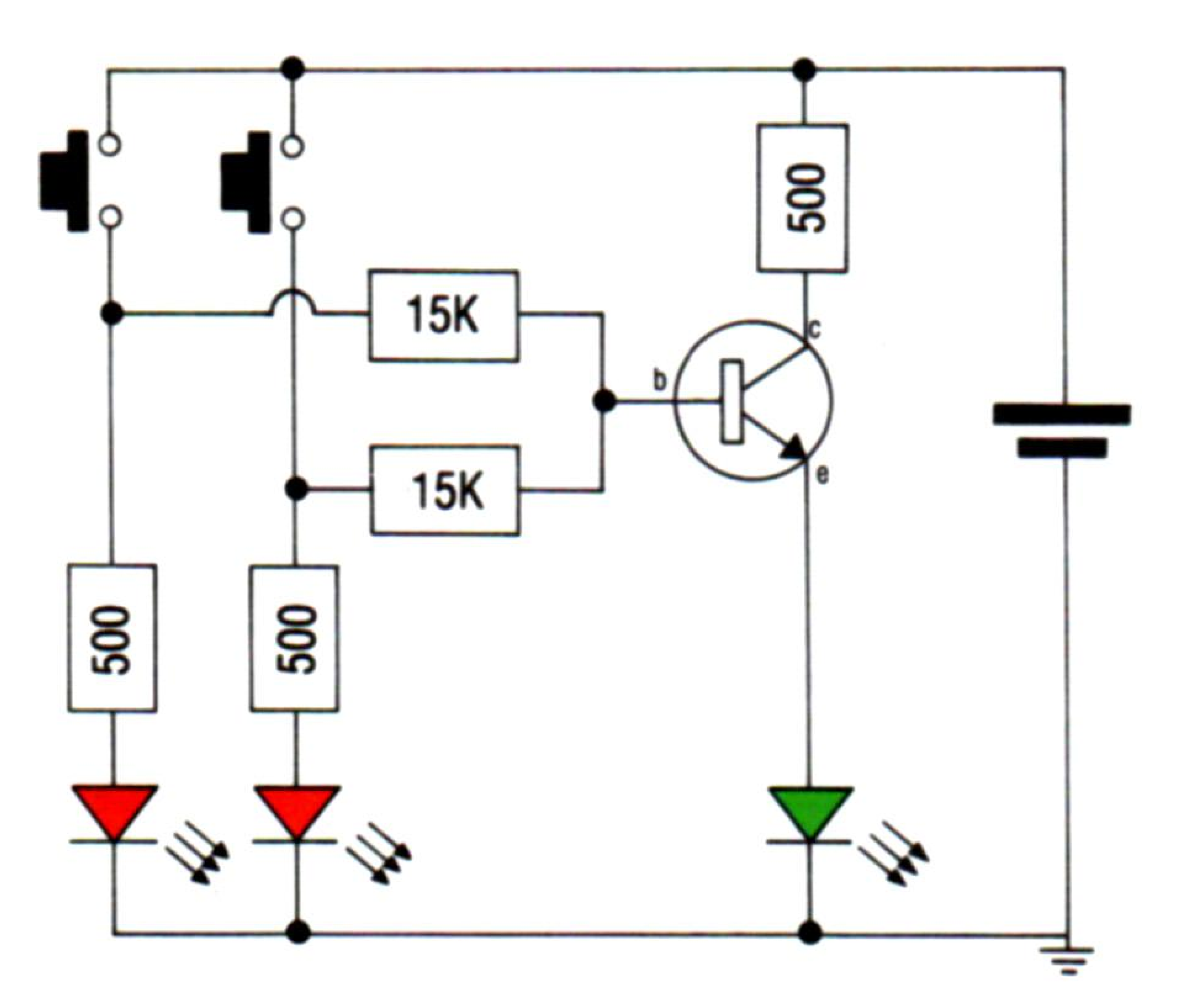




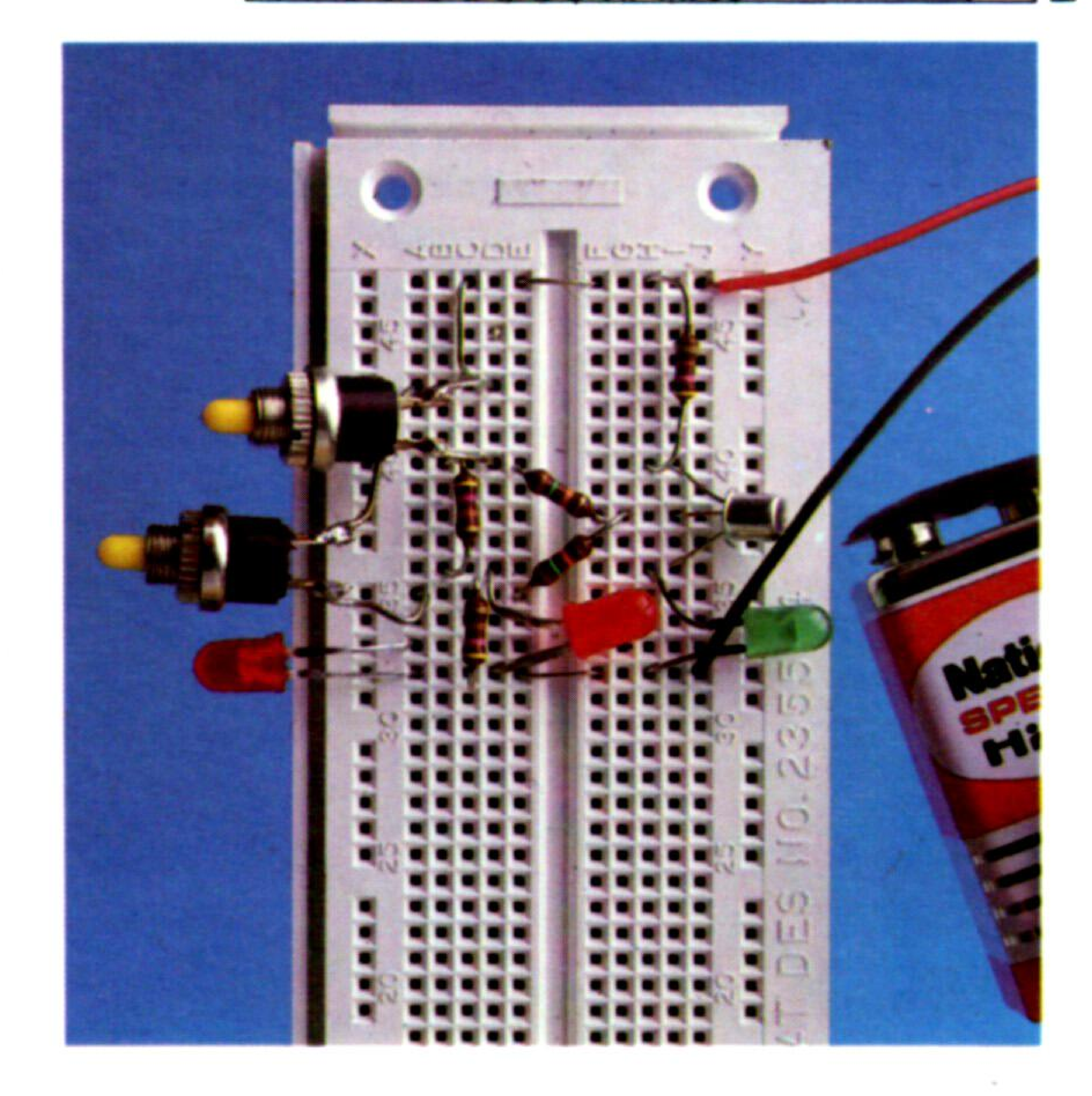
La puerta NOT
Ésta es la puerta lógica más sencilla, con una única entrada (indicada mediante el LED rojo y el interruptor) y una única salida (el LED verde). Con el interruptor abierto, no puede fluir ninguna corriente a través de la base hacia el emisor del transistor. Esto produce una gran resistencia en el colector y, como resultado, el flujo de corriente toma la ruta alternativa a través del LED verde. Por consiguiente, cuando no se aprieta el botón (o sea, cuando la entrada es 0), el LED verde se ilumina (la salida es 1). Al apretar el botón, se alimenta la base del transistor con una corriente, eliminando la resistencia proveniente del colector. Ahora la corriente fluye a través del transistor, evitando el LED. De modo que cuando se pulsa el botón (entrada 1) el LED no se ilumina (salida 0)

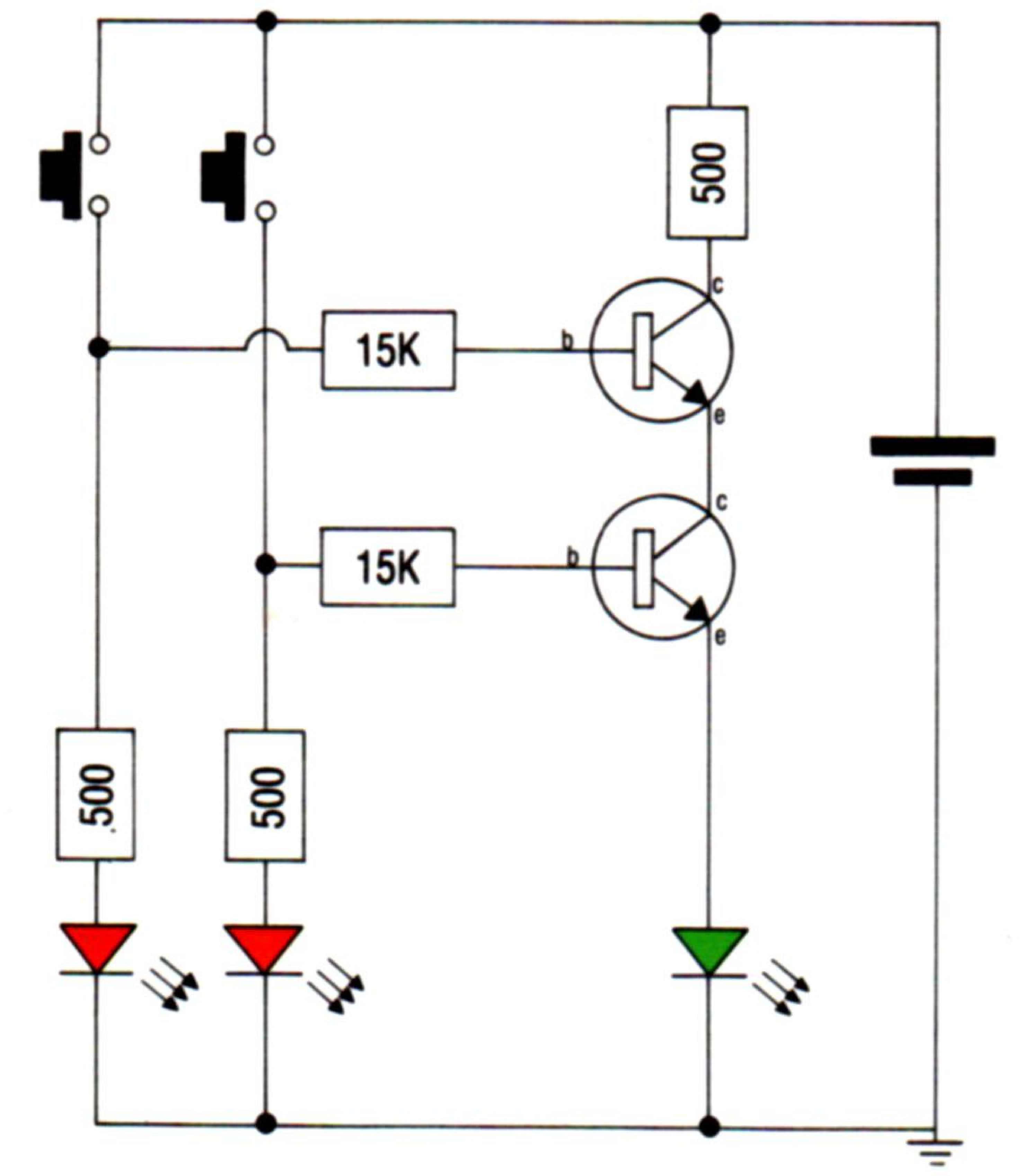


Kevin Jones

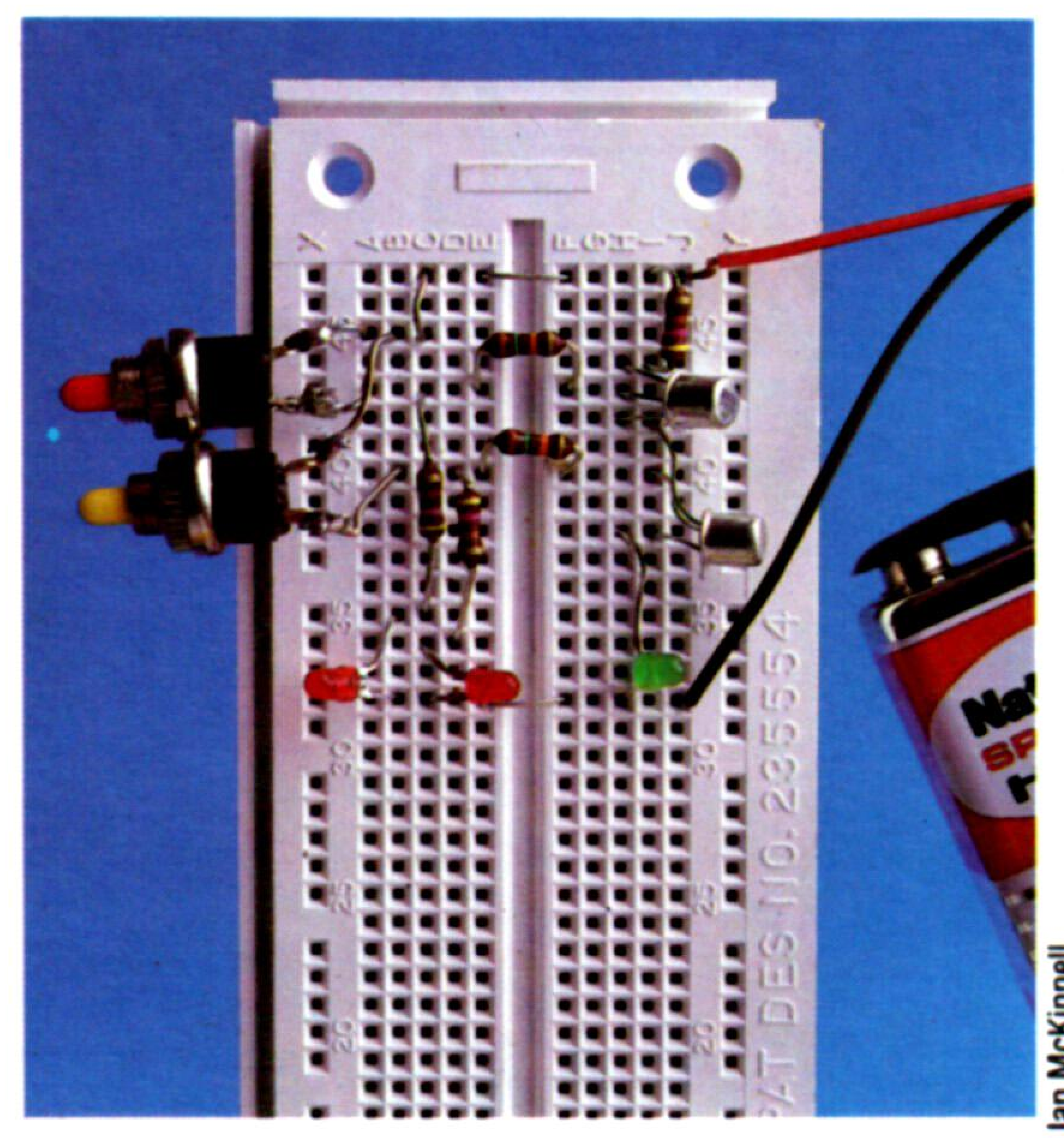


La puerta OR
El circuito utilizado para la puerta NOT se puede adaptar fácilmente para convertirlo en una puerta OR. La primera modificación consiste en colocar la salida LED de modo tal que la active una corriente que pase *a través* del transistor. En una puerta OR existen dos entradas, cada una de ellas con interruptor y LED. Cuando alguno de los interruptores alimenta una señal a la base, conmuta el transistor para permitir que la corriente atraviese el LED





La puerta AND
Para crear una puerta AND necesitamos dos transistores en el trayecto hacia la salida, cada uno con su propia entrada. Sólo cuando se pulsen ambos interruptores (es decir, cuando las dos entradas sean 1) se abrirán los dos transistores, permitiendo que la corriente fluya a través del LED



Ian McKinnell

Cambiar de línea

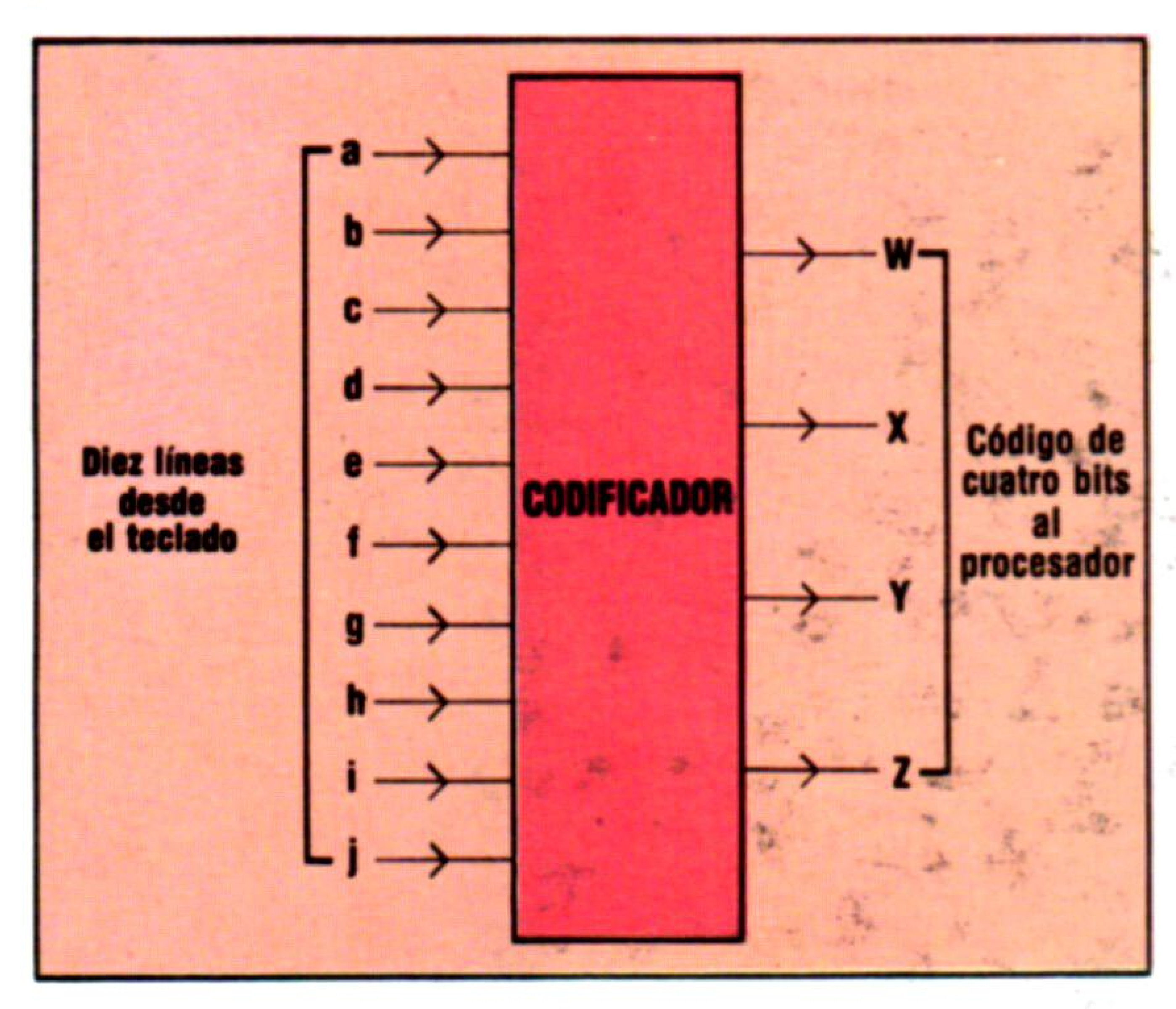
Analicemos los circuitos codificadores y decodificadores, que traducen las instrucciones a señales eléctricas y viceversa

La CPU de un ordenador personal realiza su función mediante el envío de instrucciones en forma de impulsos eléctricos, canales inferiores de comunicación con dispositivos internos o periféricos.

El procesador envía instrucciones tanto a dispositivos internos (el acumulador, la ALU, etc.) como a equipos periféricos (p. ej., una impresora). Con frecuencia se suele reducir el número de líneas utilizadas por un dispositivo periférico que proporciona al procesador una entrada simplificada. Esto se denomina *codificación*. Un ejemplo de codificador es un circuito utilizado en conjunción con un teclado. Éste podría tener 64 líneas de salida, una de las cuales produce una senal cuando se pulsa la tecla correspondiente. Como por lo general se pulsa una sola tecla cada vez, cada una de las 64 posibles señales de salida se puede codificar como un número binario de seis bits ($2^6=64$). Esto significa que sólo se requieren seis líneas para transportar hasta el procesador la información relativa a cuál de las teclas se ha pulsado. El dispositivo que convierte 64 líneas en 6 líneas es un codificador.

Para demostrar este principio vamos a considerar un teclado mucho más sencillo, de sólo 10 teclas, por ejemplo, para que el usuario trate números entre 0 y 9, ambos inclusive. Un código binario de tres bits nos daría solamente ocho (2^3) combinaciones posibles, de modo que debemos diseñar nuestro codificador con 10 líneas de entrada y cuatro líneas de salida. Como sólo una de las 10 líneas se puede activar en cada momento, la tabla de verdad para el codificador será:

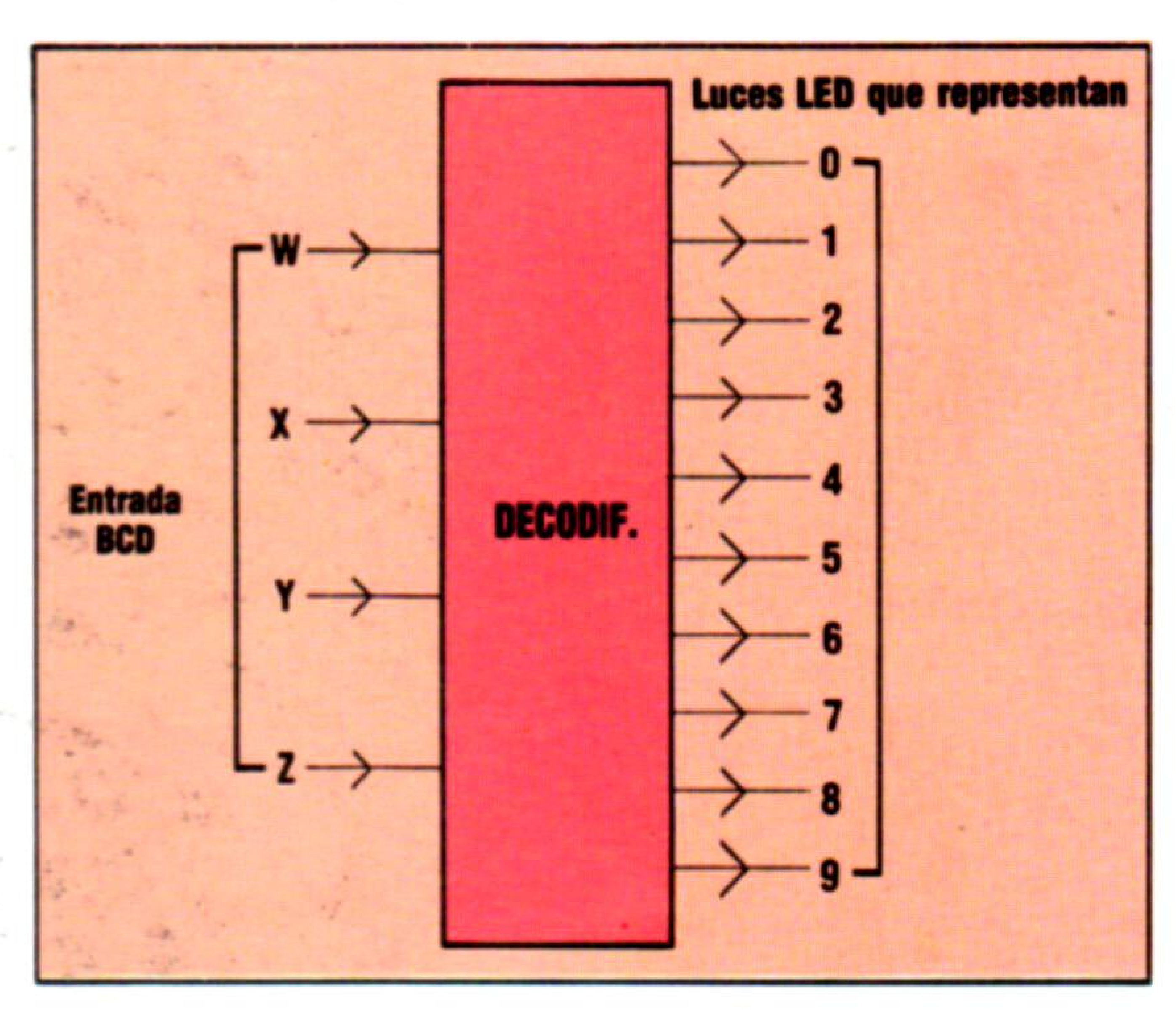
Decimal	Entradas										Salidas			
	a	b	c	d	e	f	g	h	i	j	W	X	Y	Z
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	1
2	0	0	1	0	0	0	0	0	0	0	0	0	1	0
3	0	0	0	1	0	0	0	0	0	0	0	0	1	1
4	0	0	0	0	1	0	0	0	0	0	0	1	0	0
5	0	0	0	0	0	1	0	0	0	0	0	1	0	1
6	0	0	0	0	0	0	1	0	0	0	0	1	1	0
7	0	0	0	0	0	0	0	1	0	0	0	1	1	1
8	0	0	0	0	0	0	0	0	1	0	1	0	0	0
9	0	0	0	0	0	0	0	0	0	1	1	0	0	1



Decodificar es lo contrario de codificar. En vez de aceptar un gran número de líneas de entrada para producir un número pequeño de líneas de salida, un decodificador acepta un número pequeño de entradas (por lo general, en forma de códigos binarios provenientes del procesador), y a partir de esto selecciona una de entre las numerosas líneas de salida, que controlan la actividad de un dispositivo de salida, como puede ser una impresora o un plotter x-y. Los codificadores y decodificadores también se utilizan en el control de los movimientos de la cabeza de disco y para seleccionar los canales de salida provenientes de números de dispositivo.

Diseño del decodificador

Ahora vamos a analizar cómo se puede diseñar un decodificador simple utilizando puertas AND, OR y NOT, a partir de la consideración del siguiente problema. Se requiere un decodificador para convertir códigos decimales codificados en binario (*Binary-Coded Decimal*: BCD) a una forma que encienda (coloque en *on*) una de las 10 luces LED correspondientes al valor decimal del código. Ciñéndonos al ejemplo, se trata de un circuito que producirá lo contrario que el codificador mencionado anteriormente.

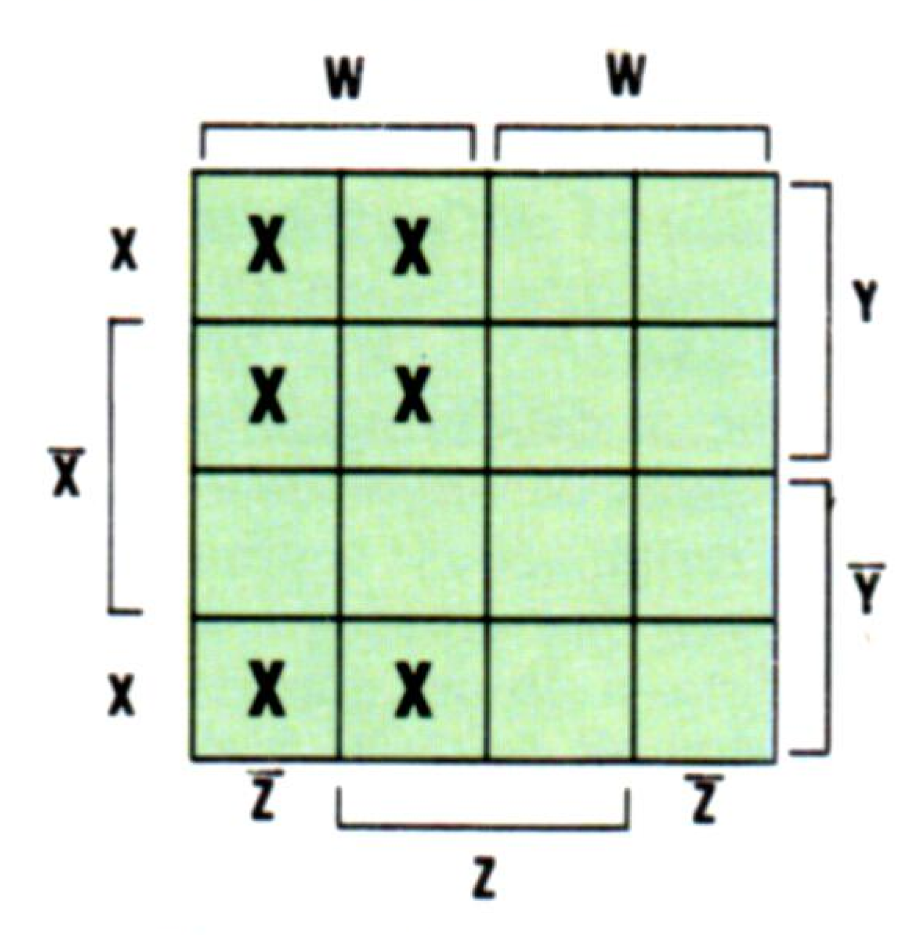


Los códigos decimales codificados en binario son las representaciones binarias de cuatro bits de los dígitos decimales de 0 a 9 y, por lo tanto, el decodificador tendrá cuatro líneas de entrada. Como se puede establecer en alto cualquier combinación de las cuatro líneas, hay 16 (2^4) posibles entradas. Nosotros sólo estamos interesados en las primeras 10 combinaciones y, por consiguiente, podemos aplicarles condiciones de "entrada invalidada" (X) a las otras seis. La tabla 1 (al margen) es la tabla de ver-

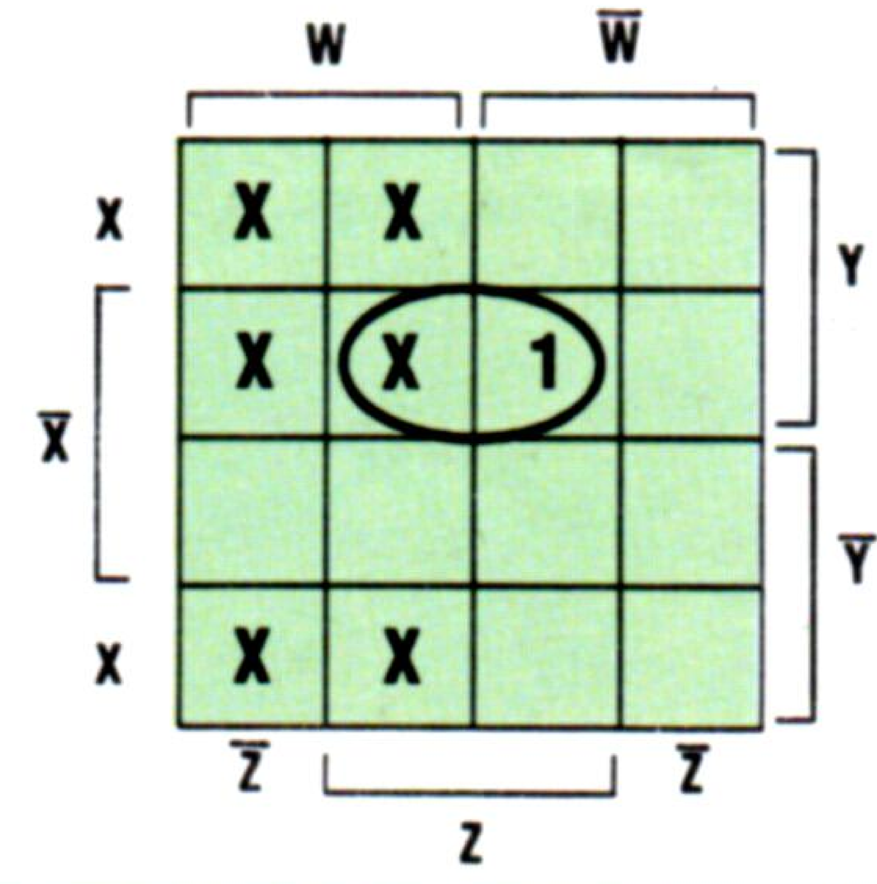


dad. Ahora bien, debemos considerar cada una de las diez salidas por separado. Parecería que la expresión booleana para cada salida debería constar de cuatro términos W, X, Y y Z, como en la tabla 2.

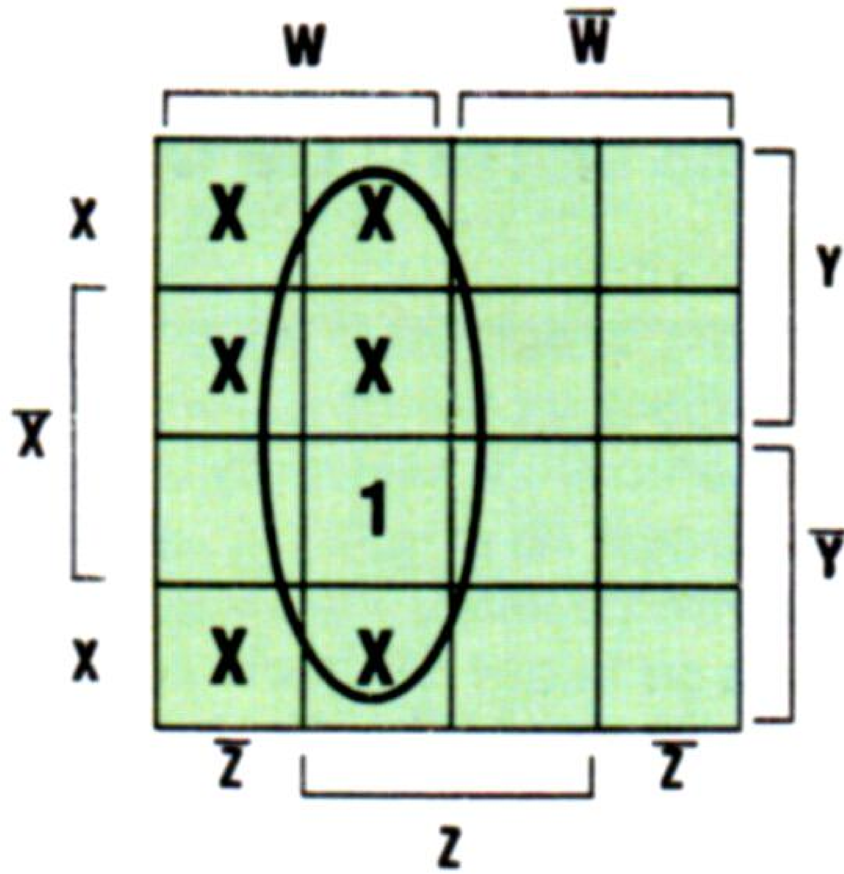
Sin embargo, podemos efectuar alguna simplificación. Como las condiciones de entrada invalidada son las mismas para cada una de las 10 salidas, podemos unir términos entre sí para simplificarlos. Abajo vemos un diagrama de Karnaugh de cuatro variables en el que están marcadas las seis condiciones de entrada invalidada.



Si ahora consideramos las salidas una a una, podremos también aquí efectuar alguna simplificación. Tomemos, por ejemplo, la salida para el dígito 3. Colocando en el diagrama la expresión booleana podemos ver que se puede dibujar un óvalo que la englobe.



De manera que la expresión para la salida se puede simplificar así: $\overline{X}.Y.Z$. Tomando la salida 9 como segundo ejemplo:



Se puede trazar otro óvalo que utiliza tres de las condiciones de entrada invalidada y representa la expresión booleana W.Z. Las demás salidas se simplifican de forma similar. Tal vez usted desee comprobar por sí mismo que los términos de salida simplificados son los reflejados en la tabla 3.

Ahora todo lo que queda por hacer es construir el diagrama del circuito a partir de las 10 expresiones booleanas. Como cada entrada se utiliza tanto en su forma normal como en su forma negativa, es más fácil construir el circuito disponiendo ocho líneas en paralelo que representen ambas modalidades. Cada una de las 10 salidas se puede entonces formar bajando las dos o tres líneas adecuadas hasta sus respectivas puertas AND. Abajo consignamos el diagrama del circuito decodificador.

Ejercicio 6

1) Se ha de diseñar un codificador de tres entradas para crear una salida 1 para las entradas 011, 101, 110 o 111. La salida debe ser cero para todas las otras combinaciones de entradas.

a) Dibuje la tabla de verdad para el codificador
b) Produzca una expresión booleana para la salida y simplifíquela
c) Dibuje el circuito codificador

Tabla 1

Entradas				Dígito BCD
W	X	Y	Z	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

Tabla 2

Dígito BCD	Expresión booleana
0	$\overline{W}.\overline{X}.\overline{Y}.\overline{Z}$
1	$\overline{W}.\overline{X}.\overline{Y}.Z$
2	$\overline{W}.\overline{X}.Y.\overline{Z}$
3	$\overline{W}.\overline{X}.Y.Z$
4	$\overline{W}.X.\overline{Y}.\overline{Z}$
5	$\overline{W}.X.\overline{Y}.Z$
6	$\overline{W}.X.Y.\overline{Z}$
7	$\overline{W}.X.Y.Z$
8	$W.\overline{X}.\overline{Y}.\overline{Z}$
9	$W.\overline{X}.\overline{Y}.Z$

Tabla 3

Dígito BCD	Expresión booleana
0	$\overline{W}.\overline{X}.\overline{Y}.\overline{Z}$
1	$\overline{W}.\overline{X}.\overline{Y}.Z$
2	$\overline{X}.Y.\overline{Z}$
3	$\overline{X}.Y.Z$
4	$X.\overline{Y}.\overline{Z}$
5	$X.\overline{Y}.Z$
6	$X.Y.\overline{Z}$
7	$X.Y.Z$
8	$W.\overline{Z}$
9	$W.Z$

CIRCUITO DECODIFICADOR

Liz Dixon

Bucles anidados

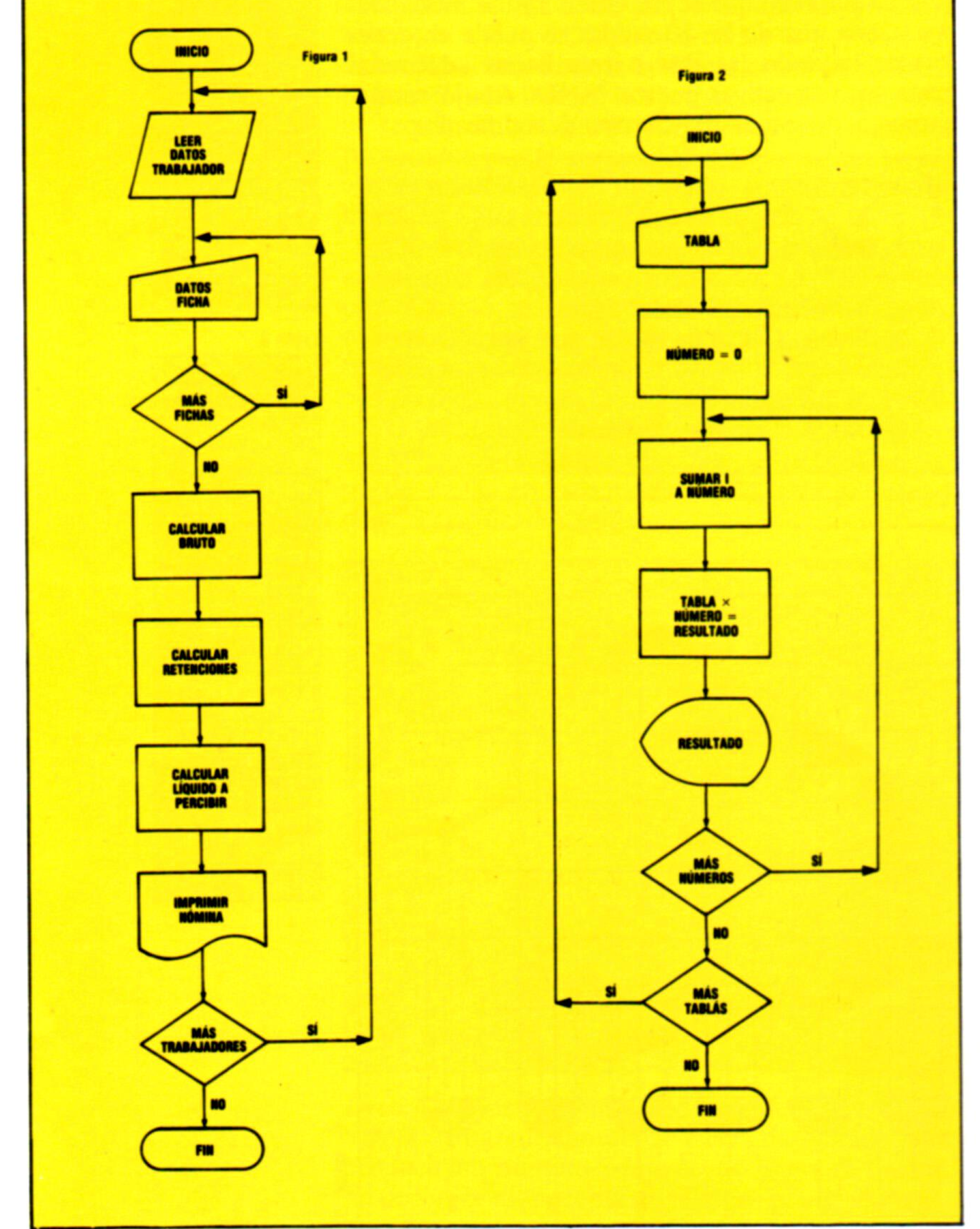
Los "bucles dentro de otros bucles" son tan útiles como delicados de tratar

Recibe el nombre de *bucle anidado* aquella parte de un diagrama en la que un bucle contiene uno o más ciclos. En este caso, las figuras nos ahorran explicaciones.

Ejemplo 1. En una posible nómina se representarán para cada uno de los trabajadores de una empresa los siguientes pasos: a) leer datos de identificación de los trabajadores; b) entrar los datos de las fichas diarias de cada trabajador; c) calcular la paga bruta, las retenciones y el líquido a percibir por el trabajo realizado; d) imprimir el correspondiente recibo (véase fig. 1).

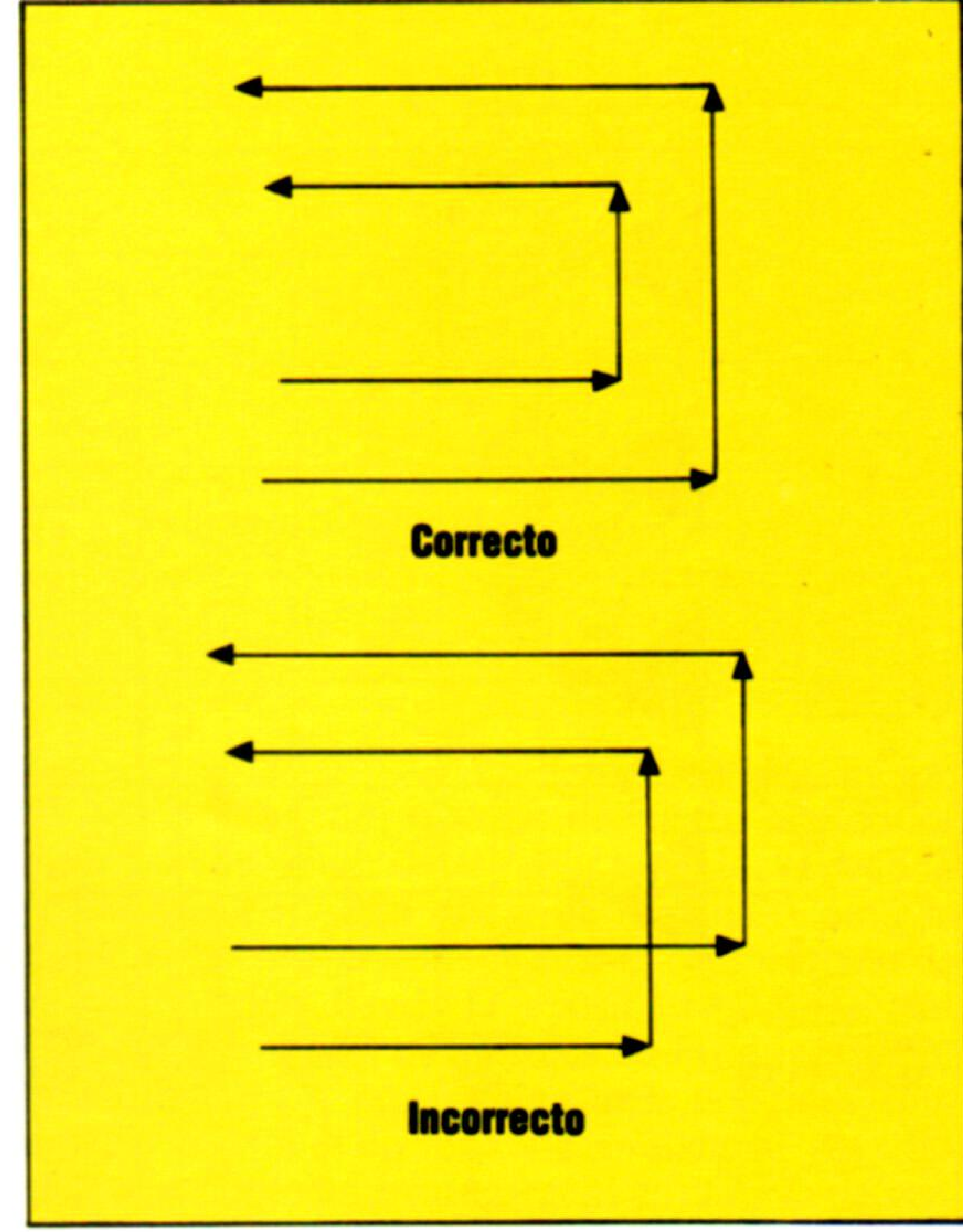
Se observa que hay un bucle general que abarca todas las operaciones que se realizan para todos y cada uno de los trabajadores de la empresa. Mientras que hay otro, interno, que se encarga de la lectura de las fichas diarias relativas a un mismo trabajador individual, ya que cada uno de ellos pudo haber trabajado una cantidad variable de días. Para salir de este bucle interno, se efectúa la pregunta de si quedan más fichas de una misma persona; de ser así, entran los datos de una nueva ficha. Cuando aquéllas se han terminado, se continúa con el resto de las operaciones diagramadas hasta formular la pregunta: ¿es ésta la última nómina por realizar? En este caso, el bucle anidado dentro del general desempeña una misión de control y lectura de las fichas diarias. Repite el ciclo una cantidad variable de veces, que depende del número de fichas, o sea de los días trabajados por cada uno de los asalariados.

Ejemplo 2. Constituye una muestra de cómo las tablas de multiplicar ya explicadas anteriormente pueden ampliarse mediante el uso de dos bucles (véase fig. 2): el *general*, que pregunta si se desea o no visualizar una siguiente tabla tras imprimir la última, y el *interno*, anidado dentro de dicho bucle de tablas, que permite confeccionar la tabla con cuantos elementos factores queramos.



Algo que se olvida

Es importante observar que los bucles anidados son aquellos que contienen totalmente uno o más bucles, y que éstos jamás deben cruzarse.



La importancia de lo estándar

Japoneses y norteamericanos se han unido con el fin de producir el MSX, que intenta proporcionar una compatibilidad entre las diversas máquinas

El Spectravideo 318 es un ordenador personal de bajo costo cuya calidad es indudablemente la misma que la del BBC Micro y el Commodore 64. Incorpora un sintetizador de sonido a tres voces, gráficos con sprites de alta resolución, una palanca de mando y una puerta para cartuchos.

El teclado se aproxima mucho a las especificaciones MSX (véase p. 621). Es de diseño similar al del Sinclair Spectrum y posee teclas de relleno a presión y plásticas del mismo estilo.

El relleno del cursor se ha convertido en una palanca de mando incorporada. Un disco sustituye a las cuatro teclas usuales con flecha y el usuario puede situar el disco en el lugar apropiado para realizar desplazamientos hacia arriba, hacia abajo, a izquierda o derecha. También se puede encajar un mango de palanca de mando y generar las pulsaciones de teclas simplemente moviendo ésta. Se puede utilizar, además, la palanca de mando para posicionar el cursor sobre los errores mientras se corrige un programa.

El BASIC de Spectravideo es el último de una línea de intérpretes escrita por la empresa Microsoft. Esta versión se aproxima mucho al BASIC MSX, que es en sí mismo una ampliación del BASIC GW utilizado en máquinas como el IBM PC. Los programas en BASIC se desarrollan fácilmente con el editor de pantalla completo y facilidades generales tales como renumeración y numeración automática de líneas. A pesar de no contar con ninguna de las nuevas órdenes "estructuradas" de que disponen muchas de las modernas versiones de BASIC, como WHILE...WEND y REPEAT...UNTIL, esta versión sí posee IF...THEN...ELSE, que es necesaria para escribir programas pulcros y eficaces. El BASIC aprovecha al máximo los gráficos de Spectravideo.

La pantalla posee una resolución razonable de 256 × 192 puntos en 16 colores, aunque grupos pequeños de puntos deben compartir los mismos colores. Esto puede parecer pobre en comparación con máquinas como el BBC Micro, pero es su nivel de control sobre la pantalla lo que permite buenos



Ian McKinnell

El teclado del Spectravideo
El teclado del 318 se aproxima mucho a las especificaciones MSX. Se trata de un diseño al estilo del Spectrum, pero el espacio dispuesto entre las teclas y la calidad de éstas hacen que resulte perfectamente utilizable. Se trata de un teclado muy completo, con todas las teclas habituales, como Control, Escape, Tab y Backspace, más cinco de función (cada una de ellas con dos funciones), una de STOP, otra de SELECT y un conjunto de teclas para edición, como INS, DEL y COPY. Mediante la pulsación de una tecla de letra conjuntamente con la tecla LEFT GRPH o RIGHT GRPH se puede disponer de una serie de formas para gráficos, como bloques, puntos y líneas. La forma que produce cada una de las teclas está claramente indicada en el teclado

gráficos y no tan sólo sus especificaciones. El 318 posee todas las ampliaciones para gráficos del BASIC Microsoft, por ejemplo, órdenes individuales para dibujar puntos, líneas, recuadros, círculos, arcos y elipses. Una orden PAINT rellena cualquier forma cerrada con un color determinado. Si el usuario desea ejercer un mayor control, instrucciones especiales VPOKE y VPEEK leen y escriben directamente en la memoria de pantalla. También hay un minilenguaje para gráficos que se emplea con la orden DRAW para crear formas y dibujos complejos.

Por medio de GET se puede "capturar" cualquier rectángulo desde la pantalla y colocarlo en una matriz, y mediante PUT hacerlo reaparecer en ella. Estas órdenes hacen que resulte fácil producir patrones regulares, animación sencilla o efectos especiales tales como invertir una imagen. Por último, el 318 posee gráficos sprites. El chip de visualización 9929 les proporciona a los programadores el recurso de diseñar sus propias formas animadas, como personas, invasores del espacio o misiles.

La instrucción ON SPRITE GOSUB le permite al usuario preparar una "trampa para incidencias". En este caso, el programa continúa normalmente; pero si dos sprites colisionaran, el BASIC saltaría a una rutina especial que se ocupara de la colisión. Esto se podría utilizar para detectar un misil que hubiera acertado a una nave espacial, por ejemplo. Al hacer uso de este recurso de interrupción, no es necesario que el programador verifique continuamente todas las incidencias que se puedan producir. En consecuencia, la programación es más sencilla y el programa se ejecuta más rápidamente.

Un buen sonido es un requisito esencial, y el chip de sonido del Spectravideo posee tres voces y una gama de efectos especiales. Puede lograr resultados notables, a pesar de que es bastante difícil alcanzar los efectos más complejos mediante el BASIC. El sonido se reproduce a través del televisor, lo que permite controlar convenientemente el volumen.

El Spectravideo posee una pequeña selección de interfaces: dos puertas para palancas de mando, una puerta para cartuchos, una puerta para cassette y un conector de ampliación. Se encuentra a la venta una interesante gama de accesorios, pero tienden a ser bastante caros. Para conectar cualquier accesorio se necesita adquirir el miniampliador. Éste le permite al usuario agregar una opción extra, por lo general una ampliación de memoria de 16 o 64 Kbytes. La ampliación mayor se consigue mediante el superampliador, que permite la adición de hasta siete accesorios utilizando un sistema de ranuras similar al del Apple II. Se puede enchufar aquí más memoria, así como interfaces para impresora, unidades de disco y modems. Si es adicto a los juegos, puede optar por un accesorio muy interesante, el adaptador para juegos Coleco.

Aunque resulte extraño tratándose de un ordenador nuevo, el 318 exige una grabadora de cassette de la misma marca. Ésta encarece el precio del sistema, si bien contribuye a proporcionarle una mayor fiabilidad y velocidad en cuanto a almacenamiento y recuperación de la información.



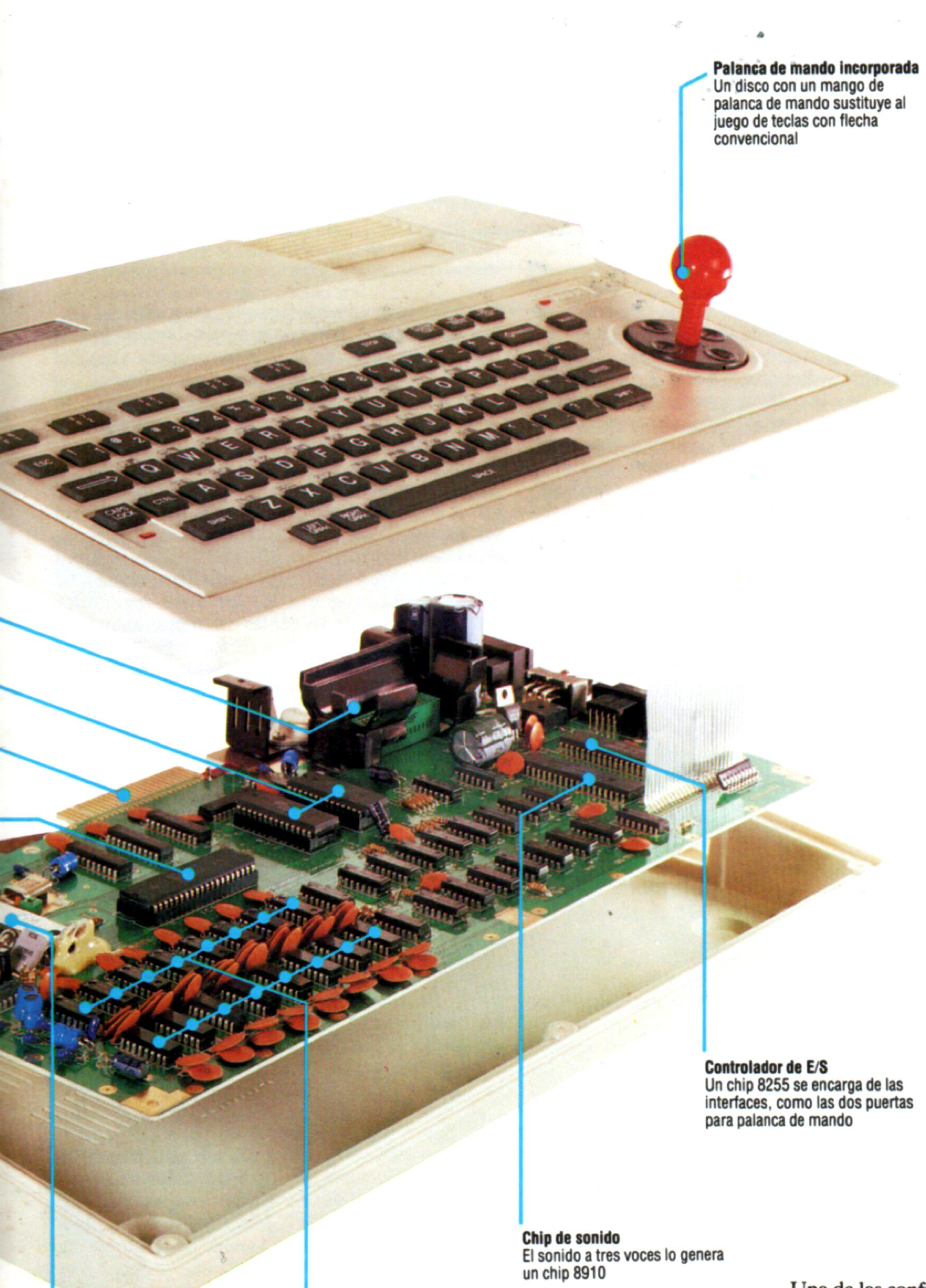
Ian McKinnell

La palanca de mando Spectravideo
Una palanca de mando incorporada sustituye a las cuatro teclas con flecha con que cuenta la mayoría de los ordenadores. Sin el mango, uno puede colocar el disco en el punto apropiado para realizar desplazamientos hacia arriba, hacia abajo, a derecha o izquierda. Con el mango colocado, al mover la palanca de mando el disco gira hasta la posición adecuada. Además, la palanca permite efectuar movimientos en diagonal presionando en dos direcciones a la vez





El hermano mayor
Una atractiva alternativa al 318 es el Spectravideo 328, versión más sofisticada del primero, con un teclado totalmente móvil, 80 Kbytes de RAM y un software para tratamiento de textos incorporado. Se pretende que sea un mejor punto de comienzo para aquellos usuarios que tengan la intención de pasar a un sistema de gestión completo



Palanca de mando incorporada
Un disco con un mango de palanca de mando sustituye al juego de teclas con flecha convencional

Controlador de E/S
Un chip 8255 se encarga de las interfaces, como las dos puertas para palanca de mando

Chip de sonido
El sonido a tres voces lo genera un chip 8910

Chip de visualización
Un disipador agregado a la ligera oculta un chip de visualización 9929

RAM
16 chips de RAM proporcionan 32 Kbytes de RAM

SPECTRAVIDEO 318

DIMENSIONES
410 × 220 × 80 mm

CPU
Z80

MEMORIA
32 K de RAM, de los cuales alrededor de 12 K están disponibles para programas en BASIC. 32 Kbytes de ROM

PANTALLA
24 filas de 40 columnas de texto, con gráficos de alta resolución de 256 × 192 a 16 colores y con facilidades para gráficos sprite. Para tareas de gestión se encuentra a la venta una opción de 80 columnas

INTERFACES
Conector de ampliación, puerta para cartuchos, dos puertas para palanca de mando

LENGUAJES DISPONIBLES
BASIC

TECLADO
Relleno plástico a presión, con teclas de función y edición y relleno para palanca de mando-cursor incorporado

DOCUMENTACION
Escasa y propensa a errores, pero la llegada de otras máquinas MSX habrá de ser un estímulo para la aparición de publicaciones independientes

VENTAJAS
Configuraciones MSX incluyendo un BASIC excelente, gráficos sprite, un teclado completo y palanca de mando incorporada. Capacidad de ampliación, en especial la posibilidad de convertirlo en un ordenador de oficina CP/M completo

DESVENTAJAS
Escasez de software. No tiene interface estándar para impresora. Requiere una grabadora de cassette exclusiva

Una de las configuraciones más atractivas del 318 en su capacidad de ampliación. Con la adición del superampliador, 64 Kbytes de RAM, la ficha de 80 columnas y una unidad de disco, el usuario puede disponer de un pequeño ordenador de oficina CP/M, con acceso al software para gestión profesional. Sin embargo, es difícil hallar programas que sean compatibles con el formato de disco Spectravideo. Esto seguirá siendo un problema mientras la máquina sea nueva y relativamente desconocida, pero se irá atenuando a medida que vaya haciéndose popular.

Estados financieros

Continuamos con nuestro estudio comparativo de tres paquetes de contabilidad, concentrándonos en sus facilidades para elaborar informes

Recuerde que los tres paquetes que estamos analizando son: *Cash trader*, de Quick Account; *Microledger*, de Lewis Ashley, y *Accountant*, de Compact Accounting Services. Además de registrar datos relativos a la compra y venta de bienes y servicios, los paquetes contables deben ser capaces de ofrecer una amplia gama de informes. Estos últimos se pueden dividir en dos categorías. Los informes de la gestión administrativa son para uso interno por parte de la empresa. Los informes financieros, o fiscales, están diseñados para que los lean organizaciones o grupos ajenos a la empresa.

Al igual que sucede con todos los sistemas computerizados, los datos básicos deben ser introducidos en el sistema antes de que se pueda elaborar con ellos algún informe. *Cash trader* y *Accountant*, por ejemplo, que no poseen facilidades para llevar archivos maestros de las cuentas de proveedores y clientes, no pueden producir los diversos informes normalmente relacionados con los sistemas de libros de compras y ventas. El *Microledger*, que posee un archivo maestro de los libros de compras y ventas, sí puede hacerlos.

El *Microledger* lleva un historial completo de transacciones para el período vigente de la cuenta de cada proveedor y cliente. También puede producir facturas que se enviarán a los clientes con el saldo adeudado y el detalle de las transacciones que lo motivan. Y puede producir avisos de remesa dineraria con destino a los proveedores (reflejando lo que se les paga). También puede proporcionar, a través de sus libros de compras y ventas, un índice de los títulos de las cuentas tanto por orden alfabético como numérico. A través del libro mayor, puede también servir una lista completa de las cuentas, un balance de comprobación y un impreso analítico ordenado según un código nominal o un código analítico, así como cada una de las cuentas.

Entre las opciones de salida impresa de que dispone el *Cash trader* figuran un resumen de los ingresos de la semana, un listado de todas las cuentas y un balance de comprobación. También puede producir extractos de la cuenta bancaria 0 de la caja, resúmenes trimestrales de las entradas o salidas del IVA, detalladas según un código IVA, y cuentas de balance.

El *Accountant* ofrece informes del libro mayor tales como un listado de las operaciones, una previsión de las cuentas (listando una cuenta determinada) y un balance de comprobación. Los informes del diario de ventas incluyen un informe agrupado de auditoría que refleja todas las transacciones pasadas para cada tipo de operación; un esquemático informe agrupado, que refleja los asientos en el libro mayor y en la cuenta de control del IVA; y un informe agrupado del IVA, que refleja los bienes y su impuesto detallados según la categoría del IVA. Los informes del diario de compras son similares, si bien lo que lucen es el IVA de entrada y no del IVA de salida, como es de suponer.

La relación de los deudores y acreedores constituye un buen ejemplo de informe administrativo para uso interno, laboriosísimo si se realiza a mano. Sin embargo, un sistema computerizado, con un programa razonablemente refinado, lo produce de forma automática con el mínimo esfuerzo adicional por parte del usuario. Se trata de un cuadro con las cantidades adeudadas o acreditadas, que se ordenan según vayan superando vencimientos de 30, 60 o 90 días. El programa lee los datos registrados comparándolos con las ventas o compras individuales y los clasifica por orden de fecha en una de las tres categorías. Cuando la cantidad se registra como pagada, se borra automáticamente de la lista de deudores y acreedores.

El informe del balance de comprobación es común a los tres sistemas. Éste imprime la descripción de cada cuenta y el total del debe y el haber actuales para cada una de ellas. Dado que los tres paquetes se basan en la contabilidad por partida doble, los pasivos y activos totales del sistema siempre deben ser iguales.

Del balance de comprobación, se pasa a elaborar el estado de pérdidas y ganancias y el balance de situación. El *Cash trader* simplifica su cuadro de pérdidas y ganancias restando sencillamente del total de los ingresos el total de los gastos. Esto puede ser suficiente para los pequeños comerciantes, pero no es un informe de pérdidas y ganancias completo y exhaustivo. Los otros dos paquetes acaban a las puertas de estos informes: llevan al usuario hasta la etapa del balance de comprobación.

No todos los informes analíticos se pueden producir con tan poco esfuerzo por parte del usuario como el balance de comprobación y el listado de deudores/acreedores. En realidad, estos informes, y algún que otro (como los listados de las operaciones) son simples "vuelcos de archivo" en los cuales a todo lo que contiene el archivo se le otorga un formato rudimentario más un título, y después se envía a la impresora. Hay cuadros mucho más selectos, cuyos criterios para la selección de datos presentan sus dificultades, con independencia del paquete que se esté utilizando.

Por ejemplo, tanto el *Accountant* como el *Microledger* poseen admirables facilidades para el análisis y la confección de presupuestos. Pero el usuario ha de poner bastante trabajo de su parte con el fin de explotar todas las ventajas que estos paquetes tienen en sus capacidades para elaborar informes de contabilidad analítica.

El *Accountant* permite que el usuario establezca

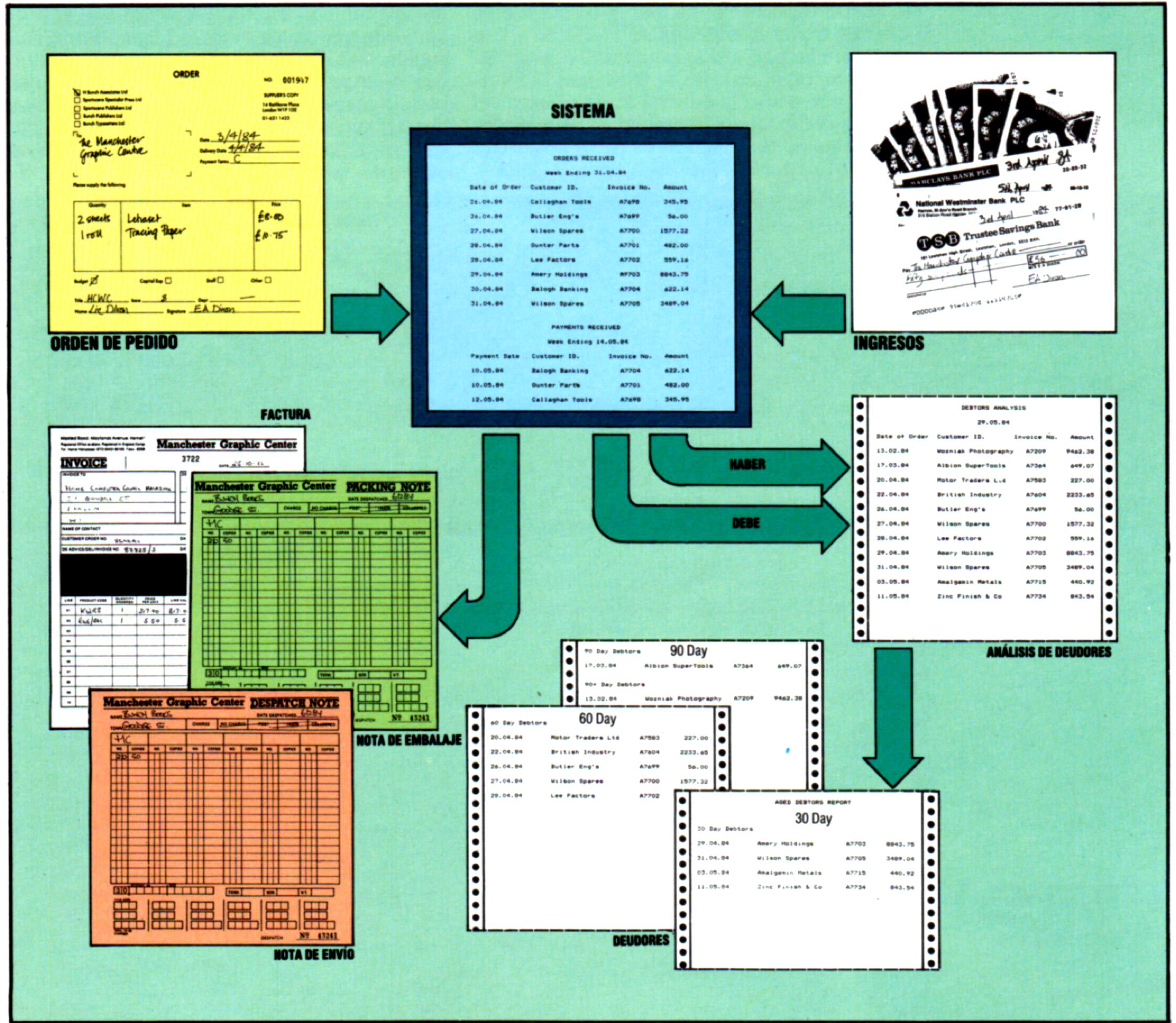
claves de grupos (o códigos) para analizar las cuentas individuales. Si, por ejemplo, usted lleva una empresa que consta de tres departamentos (supongamos, departamento de producción, de ventas y de almacén/compras), puede utilizar una clave de grupo para averiguar cuánto cuesta mantener cada departamento. Habrá costes comunes (y, por tanto, títulos de cuentas comunes) a los tres departamentos. Pero otorgando a los costes del departamento de compras la clave 01, al departamento de producción la 02 y al departamento de ventas la 03, usted puede identificar qué proporción de cada tipo de coste se debe asignar a cada uno.

Si se elige la opción "balance de comprobación formateado" del *Accountant*, obtendremos un balance de comprobación que clasifique las cuentas por su código analítico, y dé subtotales para cada departamento. Estos informes son muy valiosos para la toma de decisiones internas.

El *Microledger* posee una facilidad similar, sólo que en este caso el número del análisis está incorporado en el código de la cuenta. Recuerde que cada cuenta, ya sea ventas, compras o nominal, se identifica en este paquete mediante un número de tres dígitos. Existen dos niveles de análisis. El primer nivel corresponde al número de dos dígitos antepuesto al número de la cuenta. El segundo es un número de tres dígitos que se intercala entre el primer número de análisis y el código de la cuenta propiamente dicho (p. ej.: aa/bbb/111).

Para producir informes administrativos podemos servirnos de estos números para dos niveles de análisis de la siguiente manera: supongamos que en el libro mayor hay una sola cuenta de ingresos, pero la empresa posee cuatro vendedores. El primer número de análisis se podría utilizar para distinguir la cantidad de ingresos por ventas de las otras categorías de cuentas, mientras que los segundos tres dígitos serían más que suficientes para identificar las contribuciones efectuadas por los cuatro vendedores. Un informe que señalara los márgenes reportados para cada uno de los cuatro vendedores podría identificar a aquellos cuyo rendimiento estuviera por debajo de la media.

Informe inmediato
Un libro mayor computerizado ofrece un acceso casi instantáneo a una amplia gama de informes útiles. Aquí se muestra cómo el sistema compila información relativa a todas las operaciones y produce automáticamente una lista de los clientes y una clasificación según el vencimiento de su deuda. Un sistema más exigente dispondría de papeles preimpresos donde formularía incluso las facturas, los albaranes y los acuses de recibo



Liz Dixon

Un ángulo diferente

Las matemáticas son necesarias en numerosos programas. Si le atrae programar pero de esta disciplina sólo recuerda nebulosos conceptos, esta serie de artículos le será muy útil

¿Cuál es el tipo de matemáticas que realmente necesita usar el programador? Eso depende, y no es sorprendente, de la clase de programa que esté escribiendo. Las versiones de BASIC que se proporcionan con la mayoría de los ordenadores poseen numerosas sentencias y funciones incorporadas para manipular gráficos en pantalla (PLOT, CIRCLE, FILL, LINE, COLOUR, INK, PAPER, etc.), de modo que los problemas que supone trasladar y hacer rotar figuras sencillas en la pantalla no son muy graves. Aun así, existen ocasiones en las que se necesitarán las funciones trigonométricas de seno, coseno y tangente, para las que, afortunadamente, el BASIC dispone de un buen conjunto de facilidades.

Cuando se pasa a la estadística, sin embargo, el BASIC nos decepciona. La mayoría de las versiones del lenguaje no tienen funciones estadísticas incorporadas que ayuden a la manipulación de datos.

Si escribe guías para juegos o digitación en las que son importantes los tiempos de respuesta del usuario del programa, la mayoría de las versiones de BASIC volverán a decepcionarlo. Sencillamente, no le proporcionan al programador funciones de tiempo fiables. Éstas son las tres áreas principales (trigonometría, estadística y tiempos) que vamos a analizar en esta serie de capítulos dedicados a las matemáticas elementales.

Los estudiantes de bachillerato o formación profesional con frecuencia se preguntan cuál es la importancia que pueda tener la trigonometría en el mundo real. La respuesta considera el hecho de que la "trigo" (como la suelen llamar) es el nexo entre la geometría euclidiana, que trata con puntos, líneas y curvas, y el álgebra, que permite llegar a soluciones matemáticas con valores conocidos manipulando valores desconocidos, o incógnitas. To-

2.° CUADRANTE

$$\text{COS R} = \frac{\text{valor x negativo}}{\text{hipotenusa}}$$

$$\text{SEN R} = \frac{\text{valor y positivo}}{\text{hipotenusa}}$$

1.er CUADRANTE

$$\text{SEN Q} = \frac{\text{valor y positivo}}{\text{hipotenusa}}$$

$$\text{COS Q} = \frac{\text{valor x positivo}}{\text{hipotenusa}}$$

eje y
Q (+x, +y)
R (−x, + y)
P (a,p)
CIRCUNFERENCIA UNIDAD
HIPOTENUSA
OPUESTO
ZETA
θ
ÁNGULO RECTO
ORIGEN O
ADYACENTE
A (a,o)
eje x
S (−x, −y)
T (+x, −y)

3.er CUADRANTE

$$\text{SEN S} = \frac{\text{valor y negativo}}{\text{hipotenusa}}$$

$$\text{COS S} = \frac{\text{valor x negativo}}{\text{hipotenusa}}$$

4.° CUADRANTE

$$\text{SEN T} = \frac{\text{valor y negativo}}{\text{hipotenusa}}$$

$$\text{COS T} = \frac{\text{valor x positivo}}{\text{hipotenusa}}$$

Líneas de la circunferencia
Las funciones seno y coseno se pueden entender mejor dibujando dentro de una circunferencia una serie de triángulos rectángulos. La línea de la base se denomina *adyacente* (forma el ángulo) y la línea vertical se denomina *opuesta* (se opone al ángulo). El seno de un ángulo es una razón o proporción: la longitud del lado opuesto dividida por la longitud de la hipotenusa, que es el lado más largo de los tres

Kevin Jones

memos la parábola, por ejemplo. Si trata de una curva que posee muchas propiedades: partiendo de ellas se pueden descubrir muchas cosas útiles "a mano", o sea, con papel cuadriculado, un transportador, una regla y un lápiz. Mucho más provechoso es, sin embargo, comprender cómo la curva puede corresponderse con la fórmula algebraica: $y = x^2$.

Esta sencilla fórmula nos permite calcular valores para cualquier punto de la curva sin tener que recurrir cada vez a dibujarla.

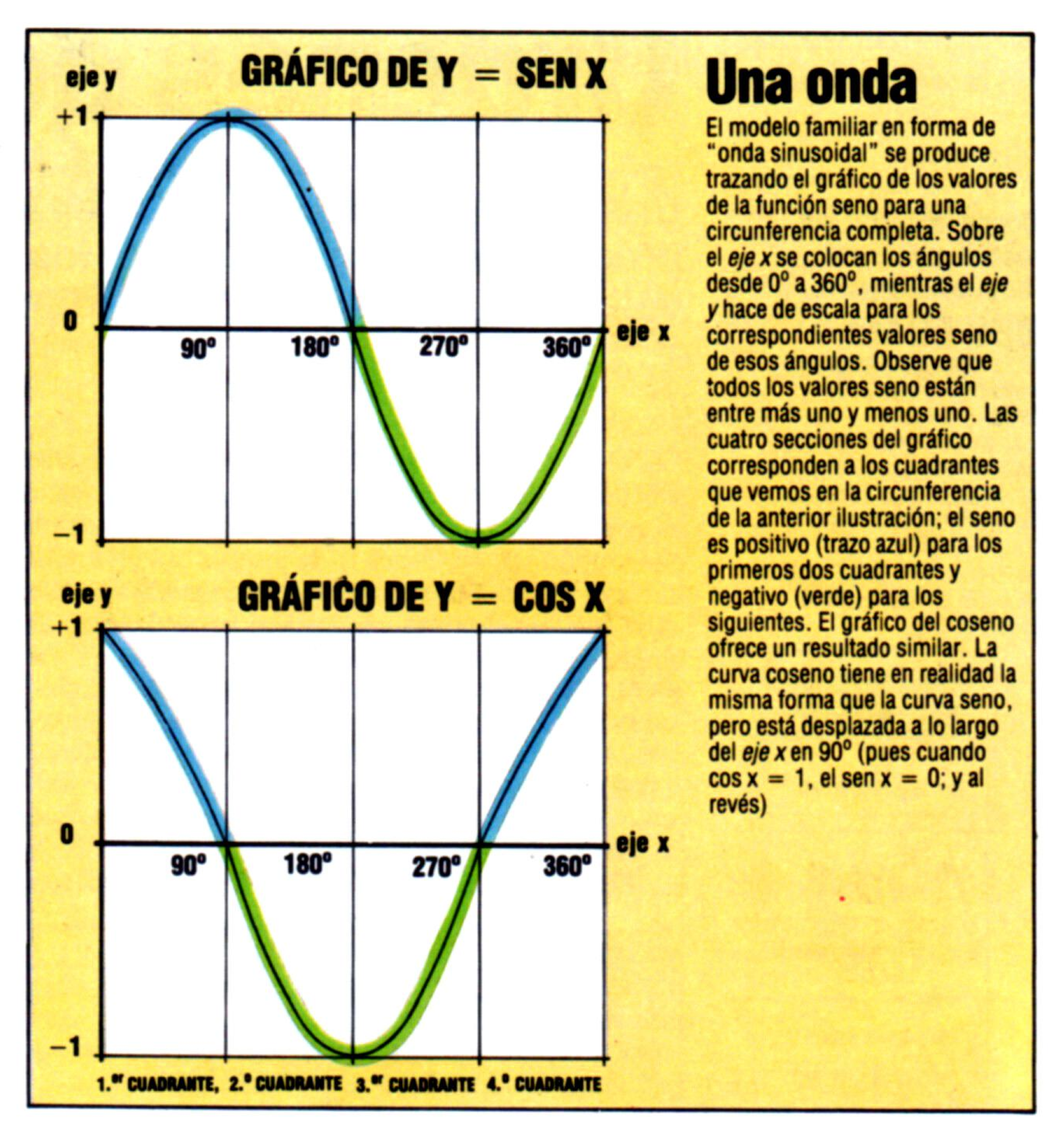
Los dos primeros capítulos de esta serie tienen como objetivo proporcionar una breve visión de los conceptos trigonométricos y ver cómo éstos se pueden fácilmente codificar en programas en BASIC. Empecemos por echar un vistazo a las funciones trigonométricas básicas *seno* y *coseno*.

Seno y coseno son dos formas de relacionar la razón de los lados en los triángulos rectángulos. El coseno y el seno también guardan una relación directa con la circunferencia. Todo triángulo rectángulo se puede trazar de modo que quepa dentro del área de una circunferencia denominada *unitaria*. Se llama de este modo porque su radio vale una "unidad"; no importa la unidad de lo que sea, porque lo que cuenta es la proporción o *razón* que ahora explicamos. En la ilustración de la izquierda vemos una línea que ha "girado" 40°. La posición inicial del giro es, por convención, el eje horizontal derecho, y la dirección del giro va "en sentido contrario a las agujas del reloj". El eje horizontal se denomina *eje de las x* y al ángulo de rotación lo llamaremos *zeta*, que se simboliza utilizando la letra griega θ. Si bajamos verticalmente un punto (p o Q o R...) de la circunferencia con una línea que llegue al *eje de las x* se obtiene un triángulo rectángulo.

El coseno de θ se define como la proporción que guarda la longitud del lado *adyacente* del triángulo (la parte que descansa a lo largo del *eje x*) con la *hipotenusa* (el radio de la circunferencia). Si usted se molesta en medirla, observará que la circunferencia del ejemplo tiene un radio de 5,22 cm y si ahora mide el lado adyacente del triángulo inscrito dentro de la misma, siendo el ángulo de 40°, veremos que vale aproximadamente 4 cm. Dividiendo 4 cm por 5,22 cm obtenemos la razón 0,76628, que es el valor aproximado del coseno para 40°. Si usted dispone de una calculadora que posea una tecla COS (también puede mirar las tablas trigonométricas, si sabe cómo hacerlo), pruebe dando entrada a 40 COS. Obtendrá como resultado 0,76604044. Esta razón se mantiene siempre, con independencia del tamaño de la circunferencia en la que esté inscrito el triángulo rectángulo.

En la circunferencia del ejemplo usted puede dibujar otros triángulos rectángulos con diferentes valores de θ. Comprobará entonces que por mucho o poco que valga θ, el valor de cos θ (cos es la abreviatura de coseno) nunca será mayor que 1 ni menor que 0, en valor absoluto. Usted mismo puede dar con la causa, con sólo pensar que si dividimos un valor menor por otro mayor el resultado nunca supera la unidad (note que el lado adyacente siempre es más pequeño que la hipotenusa).

Lo que sí puede pasar es que nos encontremos con un cos θ con valor negativo. Y esto se explica del siguiente modo: como puede ver, el giro que hace a θ más grande o más pequeño tiene por centro el llamado punto de origen 0, donde se encuentran los dos ejes *x* e *y*. Pues bien, los matemáticos



Una onda

El modelo familiar en forma de "onda sinusoidal" se produce trazando el gráfico de los valores de la función seno para una circunferencia completa. Sobre el *eje x* se colocan los ángulos desde 0° a 360°, mientras el *eje y* hace de escala para los correspondientes valores seno de esos ángulos. Observe que todos los valores seno están entre más uno y menos uno. Las cuatro secciones del gráfico corresponden a los cuadrantes que vemos en la circunferencia de la anterior ilustración; el seno es positivo (trazo azul) para los primeros dos cuadrantes y negativo (verde) para los siguientes. El gráfico del coseno ofrece un resultado similar. La curva coseno tiene en realidad la misma forma que la curva seno, pero está desplazada a lo largo del *eje x* en 90° (pues cuando cos x = 1, el sen x = 0; y al revés)

han convenido en que las mediciones que se hagan desde 0 a cualquier punto sobre el *eje x* a la derecha de 0 tengan signo positivo, y las longitudes a la izquierda sean negativas (para el *eje y*, hacia arriba tendremos los valores positivos y hacia abajo los negativos). Cuando θ rebasa los 90° (un ángulo recto), y se para, por ejemplo, en R, el lado adyacente, según lo establecido, mide −3,5 cm, valor negativo. La razón −3,5/5,22 es negativa y vale −0,670, que es lo que muy aproximadamente le daría su calculadora si digitara el coseno del ángulo 138,9°. Le resultará fácil percatarse, ayudado del dibujo, que todo ángulo entre 90° y 270° tiene coseno negativo. Por otro lado, todos los cosenos de ángulos entre 0° y 90° y entre 270° y 360° son positivos. Advierta que la hipotenusa (o radio) siempre es considerada por los especialistas en trigonometría como positiva.

Si comprendió bien lo anterior, le resultará claro cuanto le añadiremos sobre la función seno. Ésta se define como la razón lado opuesto/hipotenusa. Luego lo único que cambia es que ahora mirará el *eje y* (vertical) y no el *eje x*. Cuando θ se aproxima al grado cero, casi desaparece el lado opuesto, pero en compensación el adyacente vale tanto como la hipotenusa. Entonces, se entiende que para θ = 0° será cos θ = 1, pues lado adyacente = hipotenusa, y un valor dividido por el mismo valor siempre da la unidad. Al mismo tiempo, como el lado opuesto es cero, se sigue que sen 0° = 0. Sólo le falta verificar que para θ situado en el primero y segundo cuadrantes (valores entre 0° y 180°) el seno es positivo, mientras que los ángulos situados en el tercer y cuarto cuadrantes (valores entre 180° y 360°) tienen seno negativo: el *eje y* en este caso va hacia abajo.

Órdenes de comienzo

Una vez escrito el programa en lenguaje ensamblador, el programador debe cuidar de las "directrices ensambladoras", válidas para los dos procesadores

Dónde localizar las instrucciones en lenguaje máquina

BBC Micro

En la modalidad directa, dé entrada a:
PRINT.~PAGE
que proporciona la dirección hexa del comienzo del espacio reservado.
Después dé entrada a:
PAGE = PAGE + N
donde N, decimal, es el número de bytes que usted desea reservar



En el capítulo anterior escribimos un programa sencillo en lenguaje máquina que sumaba dos números en el acumulador y almacenaba el resultado en la memoria. No había nada asombroso en lo que hacía el programa; pero al escribirlo cubrimos muchos puntos importantes para el programador de lenguaje máquina. Volvamos ahora a mirar el programa, con las direcciones de posición incluidas, como si fuera a cargarse en $0000 y el resultado acumulado se fuera a almacenar en la dirección $0009. Las dos versiones del programa son:

Dirección de la posición	Lenguaje máquina	Lenguaje ensamblador
	Z80	
0000	A7	AND A
0001	3E 42	LD A,$42
0003	CE 07	ADC A,$07
0005	32 09 00	LD BYTE1,A
0008	C9	RET
	6502	
0000	18	CLC
0001	A9 42	LDA #$42
0003	69 07	ADC #$07
0005	8D 09 00	STA BYTE1
0008	60	RTS

Observe que la cuarta instrucción (que almacena el contenido del acumulador en $0009) de ambos programas no especifica ninguna dirección de destino en la columna de lenguaje ensamblador. En cambio, utiliza una dirección simbólica, BYTE 1. Pero en la versión de la instrucción de lenguaje máquina vemos el opcode 32 que significa "transferir lo contenido en el acumulador" seguido de 09 00, la forma *lo-hi* de los dos bytes de la dirección $0009.

Éste es otro aspecto del proceso de traducción (o ensamblaje) del lenguaje ensamblador al lenguaje máquina. Así como se usan expresiones mnemotécnicas razonablemente significativas para las instrucciones (STA y RET, p. ej., en lugar de códigos hexas como 8D y C9) porque gracias a ellas nos resulta más fácil escribir y leer los programas, del mismo modo con frecuencia utilizaremos símbolos como BYTE1 en lugar de direcciones hexas o números tan opacos como $0009. Esto se parece mucho a la inicialización de variables con constantes en un programa en BASIC, y el razonamiento es exactamente el mismo en ambos casos: el programa resulta más legible, se reduce la posibilidad de que se produzcan errores al escribir tales números y se consigue que el programa resulte mucho más fácil de modelar. Por ejemplo, si colocamos la sentencia donde a la variable se le asigna por primera vez el valor constante, haremos que el nuevo valor se utilice automáticamente a lo largo del programa, sin que precise una ulterior edición por nuestra parte.

Esto es fácil de comprender con un programa en BASIC; pero en nuestro programa en lenguaje ensamblador, ¿dónde está el equivalente de la sentencia LET BYTE1 = $0009? De momento esta instrucción no existe. Cuando lleguemos realmente a ensamblar el lenguaje máquina no debemos olvidar que esto se debe hacer por nosotros mismos. No obstante, si estuviéramos utilizando un programa ensamblador para que realizara el ensamblaje por nosotros, entonces podríamos efectuar una sentencia de asignación de este tipo al comienzo del programa (aquí proporcionamos la versión Z80 del programa, si bien estas directrices de ensamblador se podrían utilizar con ambos procesadores):

		Z80		
0000		BYTE1	EQU	$0009
0000	A7		AND	A
0001	3E 42		LD	A,$42
0003	CE 07		ADC	A,$07
0005	32 09 00		LD	BYTE1,A
0008	C9		RET	

BYTE1 está colocado en una columna propia, denominada campo de etiqueta, del cual hablaremos más adelante. En el campo del opcode se utiliza una nueva expresión mnemónica (EQU, por *equate* —igualar— o sea, "establecerlo igual a..."); y en el campo del operando se proporciona el valor que se le ha de asignar a BYTE1 (en este caso, $0009).

Es importante observar que a pesar de que EQU aparece en el campo de opcodes y tiene un aspecto mnemónico, no pertenece a la mnemotécnica del lenguaje ensamblador, ni siquiera forma parte del conjunto de instrucciones del Z80 o del 6502. Expresiones mnemotécnicas de este tipo se conocen como *pseudoops* (pseudocódigos de operación) o *directrices de ensamblador*. EQU dice al programa ensamblador: "siempre que encuentres el símbolo alfanumérico que me precede (BYTE1, en este caso) debes reemplazarlo por el valor que me sigue ($0009, en este caso)". Recuerde que cuando utilizamos un programa ensamblador lo escribimos en este lenguaje, ya sea como un archivo en disco-cinta o directamente en el teclado, y después llamamos al programa ensamblador para que lo convierta en un programa en lenguaje máquina. La salida de un ensamblador generalmente es un listado completo en este lenguaje como los que hemos estado produciendo, más el programa en lenguaje máquina que se compone sencillamente de una serie de bytes hexas. El código de lenguaje máquina se puede guardar como un archivo para uso posterior o bien cargar inmediatamente en la memoria para su ejecución.

Al realizar el ensamblaje, por nosotros, se puede hacer que el ensamblador lleve a cabo otras tareas



que hemos venido efectuando de forma manual: adosar las direcciones de posición a cada una de las líneas del programa, por ejemplo. Otro pseudo-op, ORG, hace esto por nosotros. Se le agrega al programa de este modo:

Z80		
0000		ORG $A000
A000		BYTE1 EQU $0009
A000	A7	AND A
A001	3E 42	LD A,$42
A003	CE 07	ADC A,$07
A005	32 09 00	LD BYTE1,A
A008	C9	RET

Observe que la dirección de posición que acompaña a la primera línea del programa es $0000, pero la dirección de la línea siguiente es $A000, lo que refleja el efecto de la sentencia ORG. Además, observe que en las líneas que contienen pseudo-ops no aparece ningún byte en lenguaje máquina, precisamente porque éstos no forman parte del programa y no se deben traducir a lenguaje máquina. Debido a que son configuraciones del programa ensamblador y no elementos del conjunto de instrucciones de la CPU, los pseudo-ops difieren de un programa ensamblador a otro. EQU, por ejemplo, en ocasiones se sustituye por "=" y ORG por ".=". Sin embargo, como el efecto es el mismo, nosotros vamos a seguir empleando ORG y EQU como si éstos estuvieran estandarizados.

Quizá a usted se le haya ocurrido, mientras leía acerca de las directrices para el ensamblador, que casi desde el comienzo de esta serie de lecciones hemos estado utilizando un pseudo-op; así, por ejemplo, "$", el indicador hexa. Éste no es nada más que una directriz que advierte al ensamblador de que lo que le sigue ha de tratarse como un número hexadecimal. Del mismo modo, "#", que hemos introducido en la lección anterior, es el indicador de "datos inmediatos", que significa que lo que va a continuación es una cantidad absoluta y no un indicador ni un símbolo. Si apuramos esta idea, en realidad podemos considerar al lenguaje ensamblador en sí mismo como una larga serie de pseudo-ops. La verdad es que no existe ningún impedimento para que usted se invente su propia mnemotécnica siempre que cada expresión mnemónica creada por usted se corresponda con una y sólo con una instrucción del lenguaje máquina. Un programa ensamblador muy popular para el Vic-20 hace eso: utiliza una versión no estandarizada del lenguaje assembly 6502, en parte para que se ajuste con el formato de la pantalla de 22 columnas del Vic.

En este curso nos ceñiremos, como hemos hecho hasta ahora, al empleo de las expresiones mnemotécnicas del lenguaje assembly habituales entre los fabricantes de chips, pero no viene mal recordar de cuando en cuando que eso que llamamos código de lenguaje máquina es simbólico. La CPU es indiferente a todo excepto a las indicaciones de voltaje de sus patillas de entrada-salida, de modo que cómo las describamos es sólo un asunto a convenir.

Habiendo terminado por el momento con los pseudo-ops, volvamos a inspeccionar nuestro programa para encontrar otros puntos interesantes. En particular, vamos a comparar las traducciones que hemos hecho aquí de las instrucciones LDA y LD A con sus traducciones en nuestros programas anteriores. Anteriormente habíamos escrito:

```
AD ?? ??     LDA $????          (6502)
3A ?? ??     LD A,($????)       (Z80)
```

que significa "cargar el acumulador con el contenido del byte situado en la dirección ????". La traducción del opcode "cargar el acumulador" es $AD (6502) y $3A (Z80). (Note las diferencias con la segunda línea del programa que estamos tratando.) Ésta incorpora una instrucción muy parecida: "cargar el acumulador con el valor inmediato $42". Los opcodes ahora son $A9 y $3E, para el 6502 y Z80 respectivamente; ¿por qué distintos? Quizá usted ya lo haya deducido por sí mismo. A pesar de que estamos efectuando la misma clase de operación (transferir datos al acumulador) en ambos programas, la fuente de estos datos es diferente. Por consiguiente, para la CPU se trata de operaciones distintas y poseen opcodes diferentes.

En el primer caso deseamos transferir al acumulador un dato contenido en el byte cuya dirección se indica. Nada se dice acerca del contenido de dicho byte; a la CPU simplemente se le proporcionan instrucciones para que traslade este contenido al acumulador. La CPU decodifica los tres bytes en código de lenguaje máquina, AD ?? ?? y 3A ?? ??, los cuales vienen a decirle: "interprete los dos bytes que siguen a este opcode como la dirección absoluta donde se halla el dato".

En el segundo caso, el dato a cargar en el acumulador está, sin embargo, en el byte que sigue al opcode, de modo que la CPU decodifica los dos bytes en código de lenguaje máquina, A9 42 y 3E 42, los cuales le dicen ahora: "interprete el byte que sigue a este opcode como el dato mismo". Hay algo en el opcode (A9 o 3E) que le dice a la CPU que cargue el acumulador desde el siguiente byte. Dado que su contador del programa siempre contiene la dirección de la siguiente orden a ejecutar, la CPU puede calcular la dirección del byte fuente y después llevar a cabo una sencilla operación "cargar el acumulador desde un byte direccionado".

Esto refuerza la impresión de que la mayoría de las operaciones que lleva a cabo la CPU son procedimientos muy simples y nada complicados. Un buen número de sus operaciones (que representan la quinta parte de su repertorio completo) consisten en trasladar los datos contenidos en un byte direccionado de la memoria a alguno de sus registros internos. Ésa es la tarea de todas estas operaciones "primitivas", y todo cuanto distingue a una instrucción de otra es la forma en la que se presenta la dirección del byte fuente (véase p. 464).

Investigar en profundidad las microoperaciones de la CPU es sin duda confuso al principio, pero nos será utilísimo para unificar coherentemente los numerosos detalles que se verán más tarde. Detalles que no le son necesarios si todo lo que usted desea es escribir programas en lenguaje ensamblador para conseguir mayor velocidad y eficacia. Para esto no necesita más que captar la idea, aprenderse todas las instrucciones, y recibir unos cuantos consejos sobre programación. Pero si desea tener pleno conocimiento de lo que está haciendo, necesitará algo más que limitarse a agregar otro lenguaje de programación a sus conocimientos informáticos. Descubrirá no sólo cómo trabaja su procesador, sino que esta investigación hará que el aprendizaje de otros programas en lenguaje ensamblador resulte mucho más sencillo e interesante.

Spectrum

En la modalidad directa, dé entrada a:

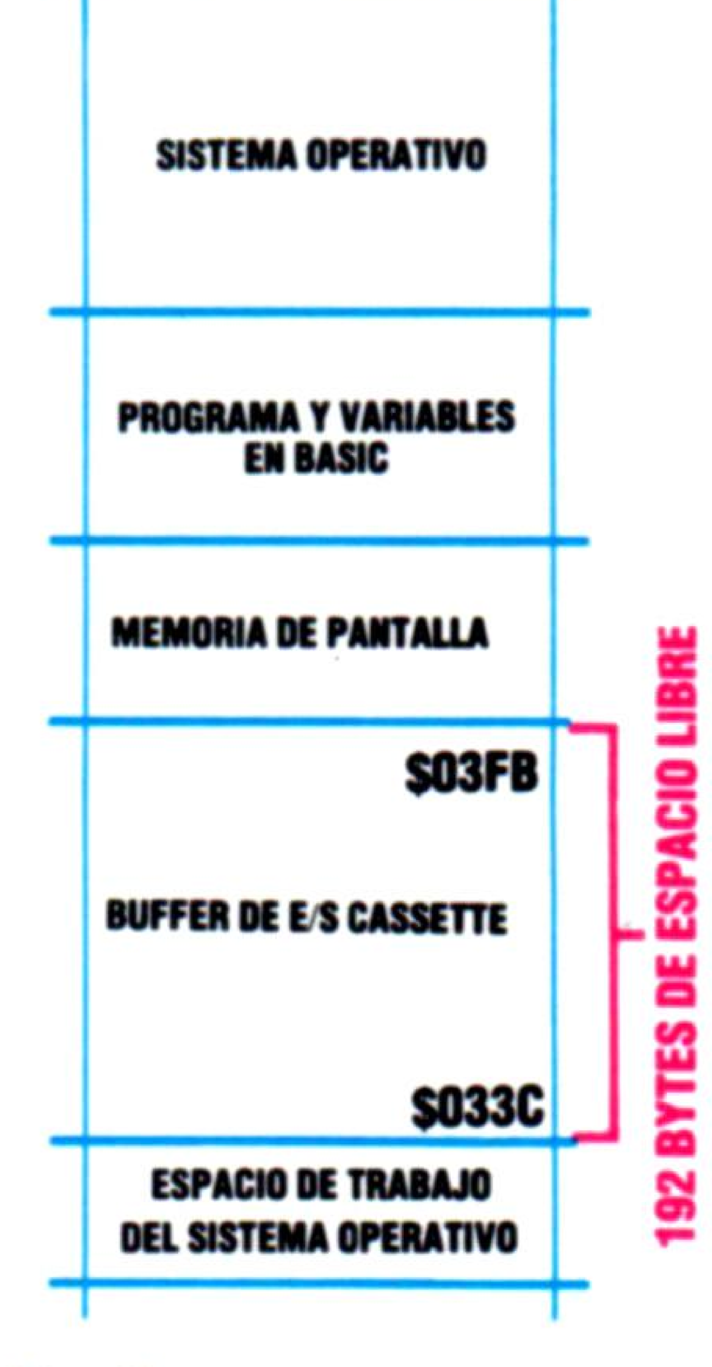
```
LET RTOP = PEEK
(23730) + 256*PEEK (23731)
LET RTOP = RTOP-N
PRINT"RTOP =  ";RTOP
```

donde N es el número de bytes que desea reservar para su programa. Su espacio reservado empieza en 1 + RTOP



Commodore 64

Utilice el buffer de cassette desde el $033C hasta $03FB



Atención

Debe ajustar estos indicadores de memoria inmediatamente después de conectar su máquina cuando no haya programa en BASIC en la memoria

Ejercicio de ensamble

1) En el recuadro de la derecha damos la versión en lenguaje ensamblador de un programa sencillo. Ensamble el programa en lenguaje máquina y determine las direcciones de posición
2) ¿Qué instrucción falta en este programa?
3) ¿Cuál es el efecto del programa en los registros y en la RAM a que se refiere?
4) ¿Qué significan las palabras "dato inmediato"? ¿Qué otros tipos de datos podría haber?
5) Si se trata a BYTE1 como una dirección, ¿en qué página de la RAM aparece?

Nota: Los valores dados en este programa sólo son teóricos: si deseara ejecutarlo, debería escoger las posiciones y los valores adecuados para su máquina

Dirección de la posición	Lenguaje máquina	Lenguaje ensamblador		
		6502		
		START	EQU	$A000
		BYTE1	EQU	$45
		BYTE2	EQU	$38
			ORG	START
			LDA	#BYTE1
			CLC	
			ADC	#BYTE1
			STA	BYTE1
			ADC	#BYTE2
			STA	BYTE2
		Z80		
		START	EQU	$A000
		BYTE1	EQU	$45
		BYTE2	EQU	$38
			ORG	START
			LD	A,BYTE1
			AND	A
			ADC	A,BYTE1
			LD	(BYTE1),A
			ADC	A,BYTE2
			LD	(BYTE2),A

El efecto de las instrucciones en lenguaje máquina

ACUMULADOR	DIRECCIÓN 1	DIRECCIÓN 2
$00	$97	$12

Instrucción 1: CARGAR al acumulador (LOAD) el byte con DIRECCIÓN (ADDRESS)1
Instrucción Z80: LDA, (ADDR1)
Instrucción 6502: LDA ADDR1

ACUMULADOR	DIRECCIÓN 1	DIRECCIÓN 2
$97	$97	$12

Instrucción 2: SUMAR (ADD) el contenido del byte con DIRECCIÓN (ADDRESS) 2
Instrucción Z80: ADC A, (ADDR2)
Instrucción 6502: ADC ADDR2

ACUMULADOR	DIRECCIÓN 1	DIRECCIÓN 2
$A9	$97	$12

Instrucción 3: ALMACENAR (STORE) el contenido del acumulador en la DIRECCIÓN (ADDRESS) 2
Instrucción Z80: LD (ADDR2),A
Instrucción 6502: STA ADDR2

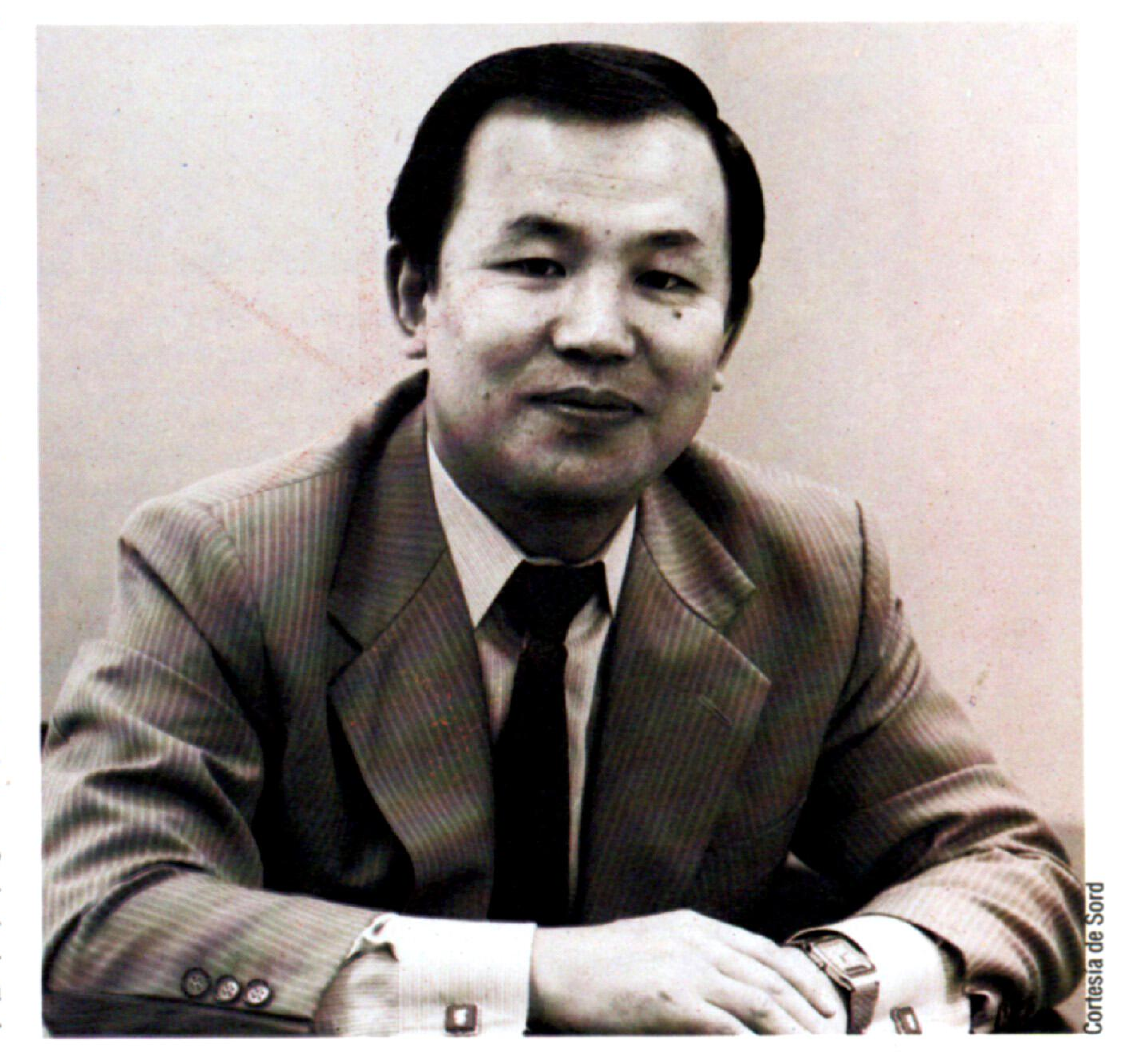
ACUMULADOR	DIRECCIÓN 1	DIRECCIÓN 2
$A9	$97	$A9

El efecto que producen las instrucciones para transferencia de datos del tipo LDA ADDR1 o LD (ADDR2),A, es siempre el de copiar el contenido de la posición fuente en la posición de destino. Por lo tanto, este contenido de la posición se copia encima en la de destino; la posición fuente no se ve afectada por la transferencia de datos



La aportación de Oriente

SORD constituye una de las más innovadoras compañías japonesas incorporadas al mercado mundial de la informática



Cortesía de Sord

Takayoshii Shiina

Probablemente la más interesante de las firmas japonesas de ordenadores sea, también, una de las menos conocidas: SORD. La mayoría de los nombres nipones familiares al oído, de Hitachi a Sony, pertenecen a enormes empresas con miles de empleados e inmensos recursos, pero SORD es una firma pequeña que sólo cuenta con unos centenares de empleados.

En nombre SORD proviene de una combinación de SOftware y haRDware. Y se trata de una elección muy adecuada, porque la empresa siempre ha prestado la misma atención tanto al desarrollo de software como a las máquinas.

Los comienzos de la empresa fueron en cierto sentido vacilantes. Su presidente, Takayoshii Shiina, abandonó la universidad para unirse a Rikei Industries, una compañía de moderado éxito de segundo rango en la Bolsa de valores de Tokio en 1967. En la misma se dedicó a reorganizar la política de comercialización de la empresa.

Hacia 1970 Shiina y un amigo suyo ya estaban preparados para fundar su propia compañía y así se formó SORD, con un capital de 650 000 yens (unas 450 000 pesetas a la cotización actual). Sin embargo, Shiina siguió trabajando para Rikei hasta el mes de diciembre de aquel año. Los primeros productos de SORD fueron un tester lógico de precio económico y algunos trabajos de programación.

Para 1971, SORD estaba empezando a llevar una cantidad razonable de negocios, entonces relacionados principalmente con la escritura de software por encargo. Hacia 1973, SORD había empezado a fabricar y hacia finales de 1974 tenía una unidad de disco flexible importada que trabajaba con una interface desarrollada por SORD. Enseguida vino el SMP-80/20, uno de los primeros ordenadores japoneses basados en el 8080 de Intel.

EL SMP-80/20 fue un producto que tuvo mucho éxito y las órdenes de pedido empezaron a llover. Pero Shiina tenía ambiciosos planes de ampliación y en 1977 formó grupo al 20 % con Toppan, una de las mayores empresas editoras del Japón. La inyección de dinero que esto significó ayudó a SORD a desarrollar un considerable apoyo de software para su creciente lista de productos para ordenadores, y para 1981 había desarrollado el PIPS.

PIPS era un paquete de programas muy avanzado para su tiempo. Fue uno de los primeros ejemplos de software integrado que se produjeron en el mundo y, más que ningún otro factor, ayudó a consolidar la posición de SORD en el mercado. PIPS combina las funciones de una hoja electrónica, un procesador de textos y una base de datos en forma tal que incluso las personas que jamás hayan usado con anterioridad un ordenador pueden aprender a utilizarlo en unas pocas horas.

El extraordinario éxito que obtuvo PIPS en Japón sólo se puede comprender si lo enmarcamos en el contexto del mercado del ordenador doméstico de aquel entonces. En Japón era una práctica generalizada y aceptada que los ordenadores se vendieran sin software de apoyo.

En Japón, el tipo de software para aplicaciones que tan familiar era en Europa y Estados Unidos para contabilidad, facturación, etc., era virtualmente desconocido. Ello se debía a dos causas. En primer lugar, la renuencia japonesa a importar algo que ellos podía hacer por sí mismos significaba que al país estaba llegando muy poco software norteamericano. En segundo lugar, los japoneses no suelen sentirse atraídos por el aprendizaje de lenguas extranjeras, ocupando una posición, en cuanto a aptitud, sólo ligeramente superior a la de los británicos. Si un manual de Digital Research sobre CP/M, por ejemplo, le resulta misterioso a un británico, a un japonés es posible que le parezca totalmente incomprensible. Los altos precios del software para aplicaciones del mercado japonés deja-



PIPS (Pan Information Processing System) es el paquete de gestión exclusivo de SORD. Se trata de un programa de uso general que se puede preparar para manipular la mayoría de las tareas de gestión contable y empresarial, desde contabilidad hasta gráficos de presentación. Demuestra un curioso enfoque japonés de la comercialización, ya que sólo se puede conseguir de forma gratuita con ciertas máquinas SORD. El PIPS podría suponer un buen motivo para adquirir un SORD, pero también limita la popularidad de un producto útil y apasionante

ron abierta una brecha para que SORD la explotara. Así fue cómo la nueva firma comenzó a desarrollar la serie M200 de ordenadores basados en el Z80 y, con posterioridad, la serie M23. Con la serie M343 se satisficieron exigencias mayores y de multiusuario. El M5 se introdujo para surtir el mercado personal de juegos, y ahora tenemos la serie M68, que incorpora tanto procesadores Z80 como Motorola 68000, los pequeños ordenadores de oficina M243 y el M285, un ordenador de 32 bits que ejecuta software VAX-11 para aplicaciones de CAD (*Computer Aided Design*: diseño auxiliado por ordenador).

Ninguna otra compañía del mundo posee una gama de productos informáticos tan amplia en el mercado; además, cada uno de ellos viene con un conjunto de software de apoyo. En lo que SORD parecería fallar es en que define a sus productos en términos de "conectar y usar". Es decir, considera sus máquinas como sistemas a pleno funcionamiento, completos con hardware y software. Los usuarios de SORD, por lo general, no pueden elegir otros productos de software para sus máquinas.

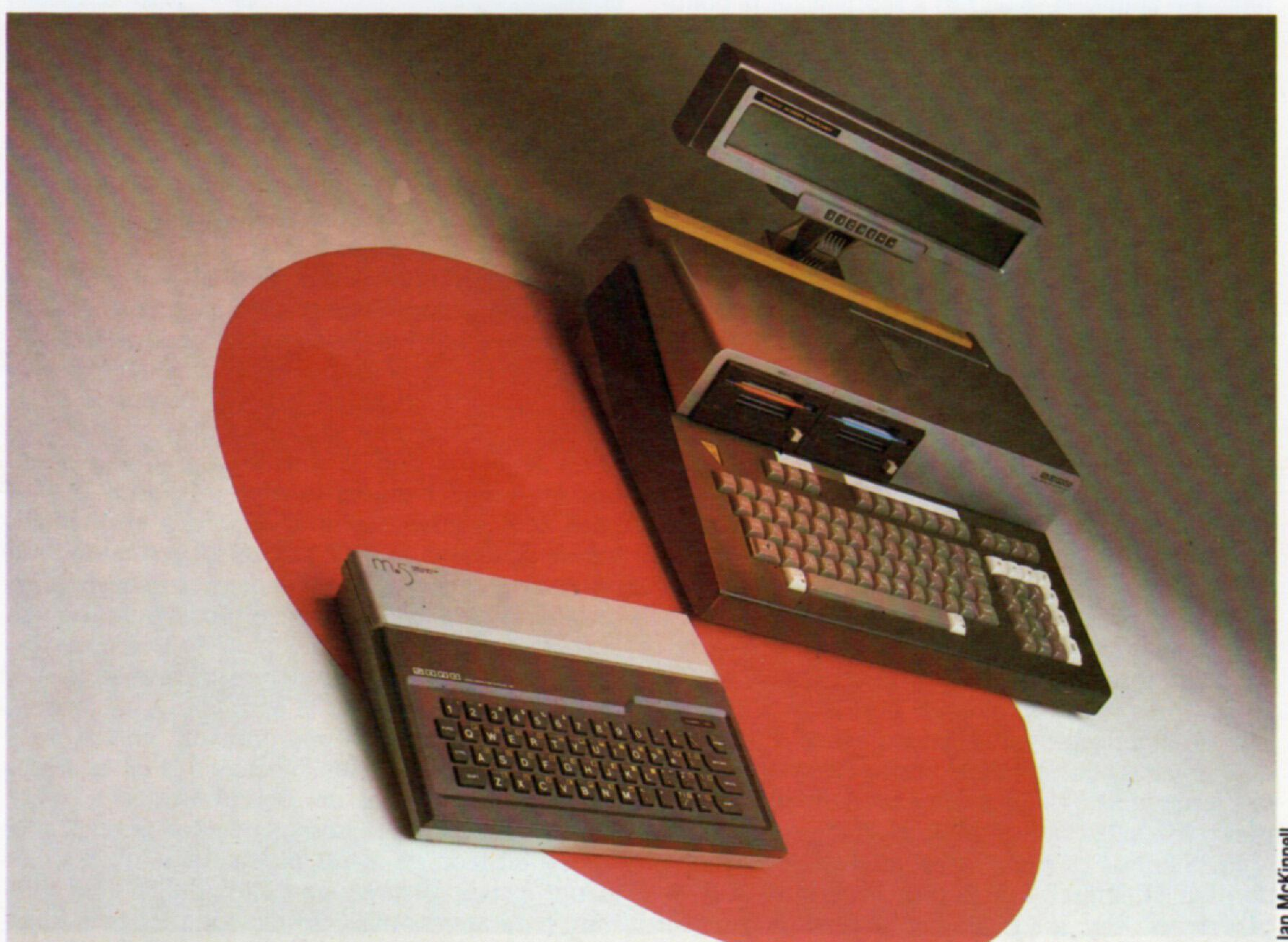
La empresa tomó algunas medidas para rectificar esta situación al proporcionar el sistema operativo SB-80 como una opción para acompañar a su serie de ordenadores basados en el Z80 M23. El SB-80 equivale al sistema operativo CP/M 2.2 de Digital Research y permite utilizar software CP/M, por primera vez, en ordenadores SORD. Otro paso que ha dado SORD en esta dirección ha sido el poner a disposición del usuario el p-system UCSD (University of California at San Diego).

Para quienes deseen seguir siendo fieles al software de SORD, la empresa ofrece varias eficaces versiones de BASIC, su propio sistema operativo, PIPS, y un procesador de textos parecido al Wang.

Aparte de los ofrecimientos de CP/M y p-system, SORD ha tendido a cubrir el campo agotado por los fabricantes de miniordenadores como DEC, ofreciendo sistemas que para su funcionamiento dependen en gran medida de software casero.

Entretanto, SORD continúa presentando innovadores diseños de hardware pero no consigue, según afirman los críticos, apoyar a los clientes con la documentación adecuada para sus productos. Durante los dos últimos años SORD ha hecho esfuerzos gigantescos por proporcionar una buena documentación, tanto para el PIPS como para su procesador de textos.

Los próximos años probablemente serán críticos para SORD, enfrentada como está al poderío industrial de las firmas japonesas de ordenadores más grandes y de la IBM. Shiina cree fervientemente en la empresa a pequeña escala y en el individualismo, y ha conseguido hacer de SORD una empresa internacional. ¿Será capaz de mantenerse en los primeros puestos en este mundo tan competitivo como es hoy el del ordenador?

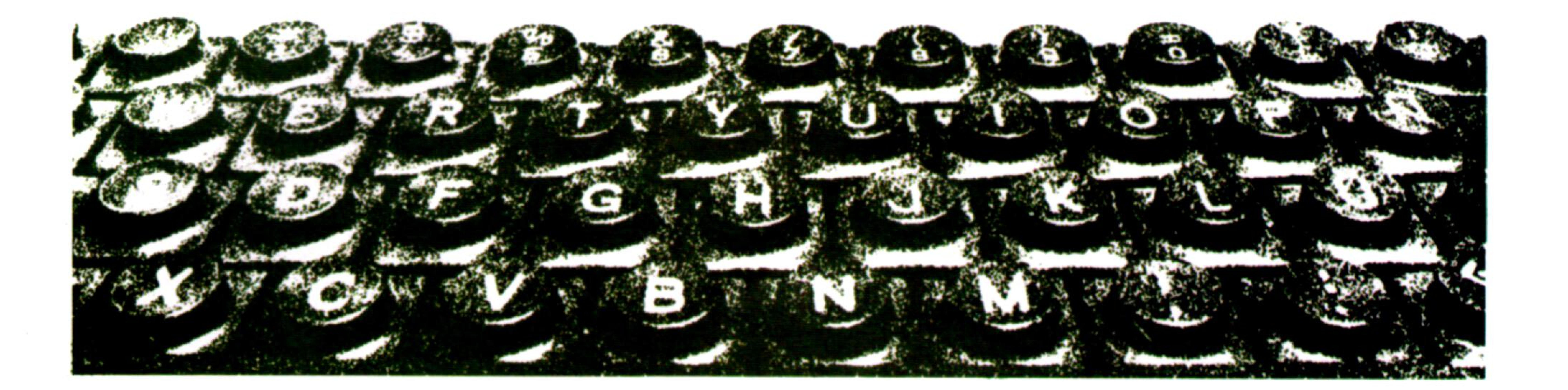


Ian McKinnell



Las opciones SORD
Las máquinas más conocidas de SORD son su ordenador personal, el M5, y su máquina portátil de oficina M23P. El M23P estableció un nuevo estándar para los portátiles mediante la utilización de discos flexibles Sony y una LCD (visualización en cristal líquido) de 80 columnas

Editorial Delta, S.A.



10032
9 788485 822836